Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *19/09/2016* **par :**

ANTHONY ZULLO

Analyse de données fonctionnelles en télédétection hyperspectrale : Application à l'étude des paysages agri-forestiers

---

JURY

HERVÉ CARDOT	Professeur	Rapporteur
STÉPHANE GIRARD	Directeur de Recherche	Rapporteur
MANUEL GRIZONNET	Ingénieur	Invité
MATHIEU FAUVEL	Maître de Conférences	Co-directeur de thèse
FRÉDÉRIC FERRATY	Professeur	Directeur de thèse

École doctorale et spécialité :

MITT : Domaine Mathématiques : Mathématiques appliquées

Unité de Recherche :

Institut de Mathématiques de Toulouse (UMR CNRS 5219)

Dynamiques et écologie des paysages agriforestiers (UMR INRA 1201)

Directeur(s) de Thèse :

Frédéric FERRATY et *Mathieu FAUVEL*

Rapporteurs :

Hervé CARDOT et *Stéphane GIRARD*

Remerciements

Je remercie Frédéric Ferraty, directeur de cette thèse, pour sa rigueur et la pertinence de ses conseils et de ses commentaires. Tu m'as fait prendre conscience de mes lacunes, mais malgré cela, tu ne m'as pas laissé tomber. J'ai mis du temps, cela n'a pas été facile, mais malgré des débuts difficiles, tu m'as finalement montré que je pouvais te faire confiance.

Je remercie également Mathieu Fauvel, co-directeur et instigateur de cette thèse, pour m'avoir fait confiance en me proposant de prolonger mon stage de Master 2 sur un sujet aussi intéressant et passionnant. J'ai vraiment apprécié la bienveillance permanente dont tu as fait preuve à mon égard du début à la fin. Merci à Frédéric et à toi pour tout le temps que vous avez consacré au suivi de mon travail depuis trois ans et demi. Merci pour votre patience.

Je remercie Michel Goulard, référent statistique et membre du comité de thèse, qui m'a apporté au quotidien son aide technique pendant le stage qui a précédé cette thèse, me permettant de mieux appréhender des logiciels tels que R ou LaTeX sous Linux.

Je remercie Marc Deconchat, directeur du laboratoire DYNAFOR, pour m'avoir acceuilli au sein de son unité. Au delà de la fonction institutionnelle, j'ai également découvert quelqu'un de profondément sympathique et humain, à l'écoute de ceux qui en éprouvent le besoin.

Je tiens à remercier tout particulierement Angel's, stagiaire DYNAFOR. D'un optimisme sans faille, tu as été pour moi au cours des derniers mois une belle rencontre, un rayon de soleil, une source d'inspiration, un inestimable soutien, un véritable ami. Personne avant toi n'a réussi à comprendre aussi rapidement quelle personne je suis. Parmi toutes tes qualités (et elles sont nombreuses !), l'une d'elles est particulièrement remarquable : tu acceptes et apprécies les gens tels qu'ils sont, au delà des apparences. Nous n'avons malheureusement pas encore eu le temps de passer un peu de temps ensemble. Je souhaite et j'espère que notre amitié saura perdurer bien au delà du cadre professionnel, notamment autour de notre passion commune : le chant. Ton amitié est le plus beau cadeau que tu pouvais me faire et j'y tiens énormément.

Je remercie les personnels du laboratoire DYNAFOR qui ont à cœur de préserver au sein de leur unité une ambiance détendue, agréable et conviviale permettant de travailler dans de bonnes conditions. Ce laboratoire est ainsi devenu pour moi un peu comme une seconde famille dont Sylvie Ladet est la maman, que je remercie d'ailleurs pour son aide quant à l'impression du présent manuscrit.

Je remercie Manuel Grizonnet, référent CNES, pour avoir fait le lien entre le laboratoire DYNAFOR et le CNES. Merci de m'avoir présenté ton service et tes collègues en compagnie d'autres doctorants. C'est à cette occasion que j'ai pu rencontrer Minh Tan, celui que je surnomme mon «frère de thèse».

Je remercie Jean-Yves Tourneret, Stéphane Girard et David Nerini pour avoir accepté de prendre part à mon comité de thèse.

Je remercie Hervé Cardot et Stéphane Girard pour avoir accepté d'être les rapporteurs de ma thèse. Merci pour les commentaires constructifs de vos rapports.

Je remercie également ma famille, notamment ma mère avec laquelle je suis très complice, mon frère avec qui j'aime passer un peu de temps, et mon père qui malgré quelques tensions est toujours là pour moi.

Je remercie l'INRA, l'ENSAT et l'IMT pour m'avoir acceuilli dans leurs locaux, ainsi que l'école doctorale MITT et l'Université Paul Sabatier.

Il me reste encore une dernière personne à remercier, celui sans qui tout cela n'aurait pas eu lieu : Philippe Vieu, je ne t'ai pas oublié. Merci de m'avoir proposé le stage de Master 2 grâce auquel j'en suis arrivé là aujourd'hui. Il est vrai que la transition entre le stage et la thèse n'a pas été facile, mais tu as eu l'intelligence de te tenir à l'écart de l'encadrement de cette thèse pour éviter les tensions et les conflits. Contrairement à ce que tu as pu craindre, mes directeurs de thèse ne m'ont pas laissé tomber, bien au contraire, ils m'ont vraiment soutenu et aidé à tenir jusqu'au bout malgré mes doutes et mes difficultés.

Je tiens à m'excuser auprès de ceux que j'aurais pu involontairement oublier de remercier ici. Les trois années et demi qui se sont écoulées depuis mon arrivée à DYNAFOR jusqu'à aujourd'hui n'ont pas toujours été faciles à vivre, tant sur le plan professionnel que personnel. Malgré la convivialité qui règne au sein du laboratoire, je ne me suis pas vraiment senti à ma place. Certains d'entre vous ont probablement dû se demander pourquoi j'avais tendance à me tenir à l'écart. Je ne souhaitais tout simplement pas imposer ma présence dans une période de ma vie personnelle où je ne me sens pas vraiment bien. Très peu de personnes de mon entourage savent en réalité ce que je n'arrive pas à assumer, ce pourquoi je me sens si mal dans ma peau. Concernant mon avenir professionnel, je pense que le métier d'ingénieur me correspond davantage que celui de chercheur. Je pense que peu d'entre vous le savent, mais en réalité, mon véritable rêve serait d'embrasser une carrière de chanteur. Malgré tout, je suis conscient de la difficulté que représente un engagement dans cette voie, c'est pour cela que je garde la tête dans les étoiles mais toujours les pieds sur terre. Je verrai bien quelles seront les opportunités qui se présenteront à moi, il ne tient qu'à moi de les saisir. Malgré tout le pessimisme qui me caractérise, je ne peux m'empêcher de me dire que dans toute expérience, même difficile, il y a toujours du positif à tirer, et celle-ci ne fait pas exception. J'ai quand même fait de gros progrès en anglais, mais pas seulement, j'ai aussi eu l'occasion de voyager un peu. Au cours de ces trois ans et demi, j'ai fait quelques belles rencontres, d'autres malheureusement plus décevantes. Malgré tout, merci pour cette aventure humaine enrichissante.

Merci à tous ceux qui ne m'ont pas laissé tomber, à tous ceux qui, de près ou de loin, m'ont supporté et soutenu jusqu'au bout. Mais il est temps pour moi de tourner la page...

Anthony

Table des matières

Cofinancement

Cette thèse a été cofinancée par le CNES (Centre National d'Études Spatiales) et la région Midi-Pyrénées.

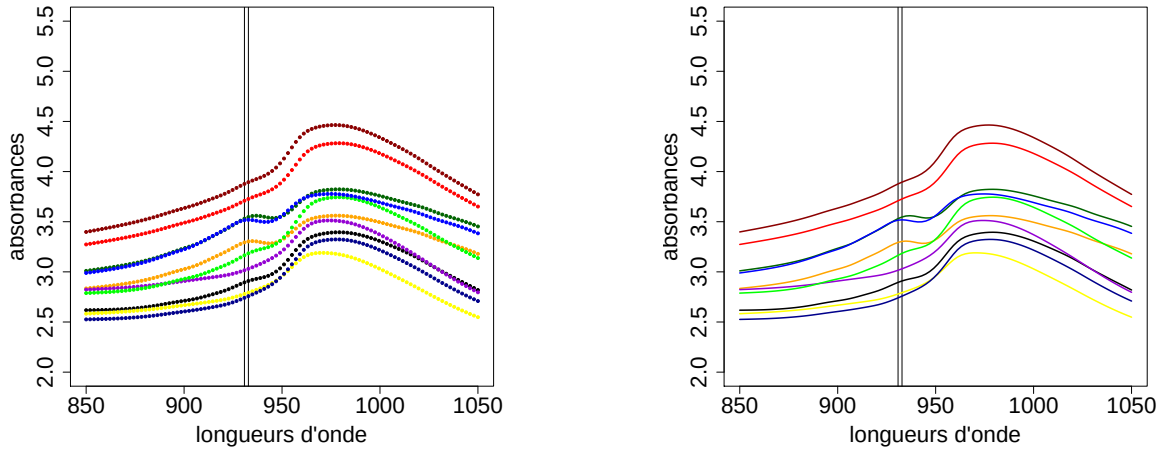
Chapitre 1

Introduction

Dans cette introduction, nous nous intéressons d'abord aux notions élémentaires de l'analyse de données fonctionnelles. Nous présenterons ensuite les données hyperspectrales et comment nous les avons étudiées à l'aide de méthodes issues de l'analyse de données fonctionnelles. Cette introduction se conclut par les contributions principales de ces travaux.

1.1 Eléments introductifs sur l'analyse de données fonctionnelles

D'un point de vue probabiliste, on appelle communément «données fonctionnelles» toute réalisation d'une variable aléatoire à valeurs dans un espace F de dimension infinie. L'exemple le plus simple pour F correspond à un espace de fonctions (plus ou moins régulières selon les contextes) de $\mathbb{R}$ dans $\mathbb{R}$. Pour cet espace F particulier, une donnée fonctionnelle se présente sous la forme d'une courbe. Un individu est alors directement associé à un objet fonctionnel, en complément d'éventuelles caractéristiques plus standards. Un jeu de données fonctionnelles contient ainsi une collection de courbes, voire un ensemble d'objets plus complexes. En médecine par exemple, l'usage de l'imagerie permet d'obtenir des collections d'images, voire des collections de surfaces pour étudier l'évolution de maladies dégénératives, de tumeurs, de déformations du système vasculaire, ... Ce sont à la fois les progrès techniques, l'augmentation des capacités de stockage de l'information, l'amélioration de l'outil informatique et de ses capacités de traitement, la multiplication des systèmes de monitoring, le perfectionnement des capteurs, qui ont favorisé l'émergence de ce type de données. Elles sont aujourd'hui couramment utilisées dans de nombreux domaines d'application : astronomie, biologie, climatologie, écologie, chimie, économie, médecine, sciences de l'ingénieur, ... Le statisticien a développé depuis une vingtaine d'années des méthodologies statistiques adaptées à ces données fonctionnelles d'un nouveau genre. L'analyse fonctionnelle des données ou Functional Data Analysis (FDA) recouvre par définition l'ensemble des méthodes statistiques impliquant des données fonctionnelles. Il existe une littérature assez dense autour de l'analyse de données fonctionnelles qui témoigne d'un engouement important de la communauté internationale de statistique pour ce sujet. Initalement introduite dans [173] pour des modèles linéaires, la notion de données fonctionnelles a été étendue dans [87] pour les modèles non-paramétriques, puis adaptée à diverses situations dans [84]. Cette notion a ensuite été étudiée à travers les processus linéaires [16]. [111] s'est plus particulièrement intéressé à l'inférence pour des données fonctionnelles. Des fondements théoriques autour des opérateurs linéaires fonctionnels sont disponibles dans [113].



(a) (b)

FIGURE 1.1 – Représentation de 10 spectres d'absorbance (données *Pork*) : (a) courbes discrétisées (représentation vectorielle) et (b) version continue (représentation fonctionnelle).

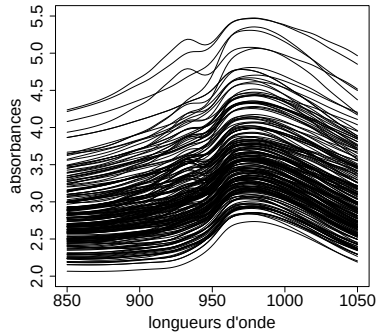


FIGURE 1.2 – Courbes représentant les 215 profils spectraux (données *Pork*).

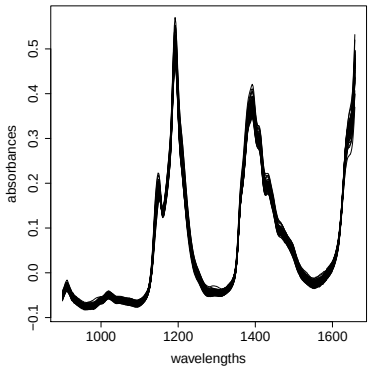


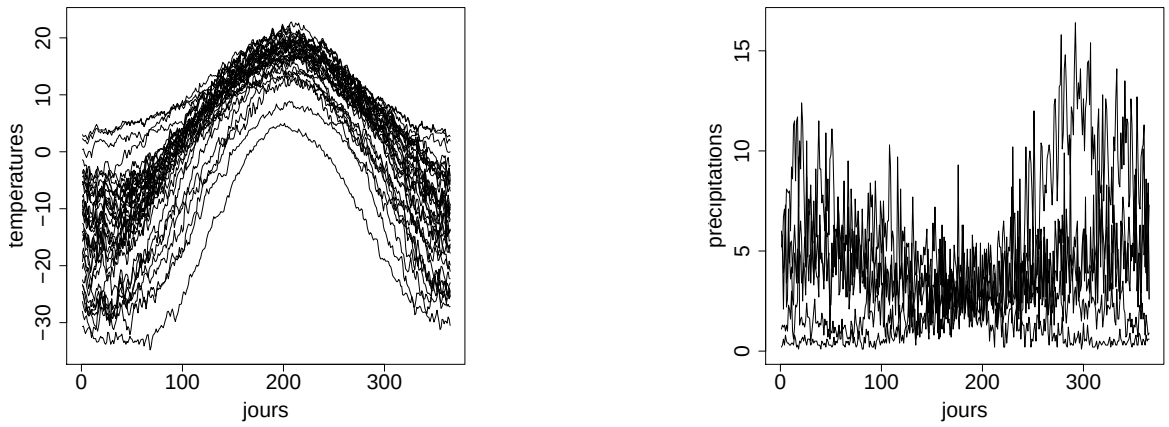
FIGURE 1.3 – Courbes représentant les 60 profils spectraux (données *Gasoline*).

1.1.1 Quelques exemples de données fonctionnelles

Afin d'illustrer notre propos, nous présentons ici plusieurs exemples de données fonctionnelles provenant de divers domaines d'application et largement utilisées dans la littérature statistique.

Le premier exemple (données *Pork* [15]) utilise la spectrométrie infrarouge, une technologie non destructrice permettant de détecter et mesurer les concentrations de nombreux composés chimiques. Les données *Pork* ont été obtenues en mesurant l'absorbance de 215 morceaux de viande de porc finement hachés pour 100 longueurs d'onde régulièrement espacées $\lambda_1, \ldots, \lambda_{100}$ de 850 à 1050 nm. Chaque spectre est donc représenté par un vecteur $X_i = (\mathcal{X}_i(\lambda_1), \ldots, \mathcal{X}_i(\lambda_{100}))$, discrétisation d'un objet de nature fonctionnelle $\mathcal{X}_i = \{\mathcal{X}_i(\lambda), \lambda \in [850\ ; 1050]\}$. La dualité de ces deux représentations est illustrée par la figure 1.1, discrétisée pour l'approche vectorielle (a) et continue pour l'approche fonctionnelle (b). La figure 1.2 montre le profil spectral de chacun de ces 215 morceaux de viande.

Un deuxième exemple de données spectrométriques (données *Gasoline* [123]) provient de mesures d'absorbance d'un faisceau lumineux décomposé en 401 longueurs d'onde régulière-



(a) (b)

FIGURE 1.4 – Courbes représentant les 35 profils annuels de température (a) et 5 profils annuels de précipitations (b) (données *Canadian Weather*).

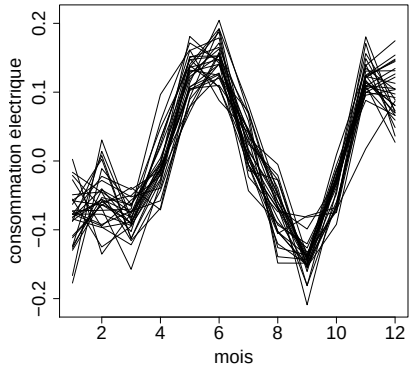


FIGURE 1.5 – Courbes représentant les 28 profils de consommation d'électricité (données *Electricity Consumption*).

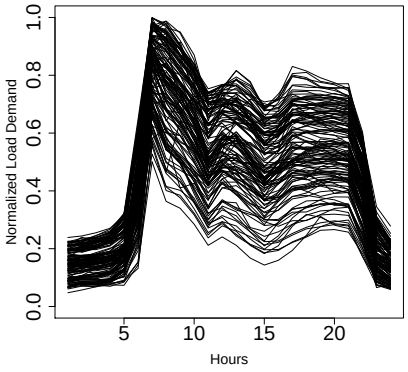


FIGURE 1.6 – Courbes représentant 135 profils de demandes de consommation en énergie (données *Heating*).

ment espacées dans la gamme du proche infrarouge (900-1700 nm), ceci pour 60 échantillons de carburant (essence). La figure 1.3 représente les 60 spectres d'absorbance ainsi obtenus (les 21 dernières longueurs d'onde ont été supprimées du graphique pour des raisons de lisibilité). De façon analogue, chaque spectre est représenté par un vecteur $X_i = (\mathcal{X}_i(\lambda_1), \ldots, \mathcal{X}_i(\lambda_{401}))$, discrétisation d'un objet de nature fonctionnelle $\mathcal{X}_i = \{\mathcal{X}_i(\lambda), \lambda \in [900\,; 1700]\}$.

Le troisième exemple (données *Canadian Weather* [173]) est composé de relevés journaliers moyens de températures et de précipitations dans 35 stations différentes du Canada entre 1960 et 1994. On obtient ainsi deux groupes de 35 courbes annuelles (Figure 1.4) où chacune d'elles représente la moyenne des relevés journaliers à une station donnée. Dans un souci de lisibilité du graphique, seuls 5 profils de précipitations ont été tracés sur les 35 que comptent ces données.

Un autre exemple (données *Electricity Consumption* [200]) provient des relevés de consommation d'électricité mensuelle des secteurs résidentiels et commerciaux des États-Unis entre janvier 1973 et février 2001. Chacune des 28 courbes (Figure 1.5) est alors composée de 12 points

représentant les relevés de chaque année, transformés par un logarithme et une dérivation.

Un dernier exemple (données *Heating* [97]) concerne les demandes de consommation d'un chauffage urbain de Turin, au nord de l'Italie. 198 courbes journalières de relevés horaires (soit 24 observations par jour) ont été collectées pour chacune des quatre périodes s'étendant d'octobre à avril entre 2001 et 2005 (Figure 1.6).

Toutes ces données ont en commun leur nature continue, caractéristique des données fonctionnelles, malgré une discrétisation plus ou moins grossière selon les cas. La plupart des méthodes statistiques multivariées n'est cependant pas adaptée au traitement de ces données fonctionnelles. En effet, contrairement aux méthodes fonctionnelles, l'utilisation de méthodes multivariées ne permet pas de prendre en compte les aspects inhérents à la nature fonctionnelle des données. Parmi les problèmes rencontrés, on note l'absence de prise en compte de la continuité, du caractère ordonné de la discrétisation et de l'apparition de colinéarité. Cette dernière particularité est illustrée par la figure 1.1, où les deux lignes verticales indiquent deux points de discrétisation consécutifs λ_j et λ_{j+1}. À partir de λ_j et λ_{j+1}, on construit deux variables dont les 215 observations fournissent les deux variables statistiques $(\mathcal{X}_1(\lambda_j), \ldots, \mathcal{X}_{215}(\lambda_j))^T$ et $(\mathcal{X}_1(\lambda_{j+1}), \ldots, \mathcal{X}_{215}(\lambda_{j+1}))^T$. Il est clair que ces deux variables sont fortement corrélées (quasiment colinéaires). Plus généralement, une approche multivariée de ces données implique qu'il y a autant de variables que de points dans la discrétisation. Or, en théorie, de par la nature continue de ces données, on peut avoir autant de points de discrétisation que l'on souhaite, la seule limitation étant d'ordre technologique. On est alors confronté à un phénomène bien connu en statistique, à savoir le «fléau de la dimension» [62]. Ce phénomène apparaît lorsque le nombre d'observations est faible devant le nombre de variables (théoriquement infini dans le cadre de données fonctionnelles) conduisant à l'isolement des observations dans un espace de grande dimension. Il semble donc légitime d'aborder le traitement de données fonctionnelles à travers le prisme des méthodes dites fonctionnelles, spécialement étudiées pour résoudre ce type de problèmes. Il existe plusieurs approches complémentaires. Une première approche consiste à décomposer les données dans une base fonctionnelle, puis d'appliquer aux coefficients ainsi obtenus des méthodes multivariées standard. On passe ainsi de données définies dans un espace fonctionnel F à des données définies dans un espace multivarié $\mathbb{R}^d$, où $d \in \mathbb{N}^*$ est la dimension de l'espace de projection [201, 216]. Une deuxième approche consiste à approximer le ou les paramètre(s) du modèle à estimer en le(s) décomposant dans une base de fonctions (Fourier, ondelettes, splines, ...). Cette approche est souvent utilisée dans un cadre d'estimation du modèle linéaire fonctionnel [143, 53]. Il existe d'autres approches fonctionnelles qui ne nécessitent aucune décomposition ni des données, ni des paramètres. C'est par exemple le cas des méthodes non-paramétriques fonctionnelles [87] basées sur un estimateur à noyau, méthodes que nous aurons l'occasion d'aborder plus en détails dans la suite de ce manuscrit. Parmi tous les types de problèmes statistiques impliquant des données fonctionnelles, nous avons choisi dans cette thèse de nous concentrer principalement sur deux d'entre eux : la régression et la classification supervisée.

1.1.2 Problèmes de régression dans un cadre fonctionnel

Dans un cadre de régression, on cherche à expliquer une variable aléatoire Y absolument continue, appelée «variable réponse», à partir d'une variable fonctionnelle $\mathcal{X}$ («variable explicative»). Le modèle de régression fonctionnelle s'écrit $Y = r(\mathcal{X}) + \varepsilon$ où r est un opérateur fonctionnel inconnu à estimer et ε est une variable aléatoire centrée dénotant l'erreur du modèle. On dispose de n couples $(\mathcal{X}_i, Y_i)_{1 \leq i \leq n}$ indépendants et identiquement distribués selon une loi inconnue du couple de variables aléatoires $(\mathcal{X}, Y)$, où $\mathcal{X}_i$ (respectivement Y_i) est la i-ème réalisation de la variable fonctionnelle $\mathcal{X}$ (respectivement de la variable réelle Y). À titre d'exemple, intéressons nous plus particulièrement aux données *Pork* et aux données *Gasoline*. À chaque

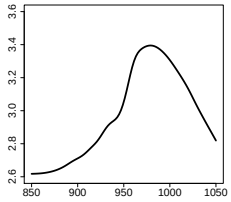
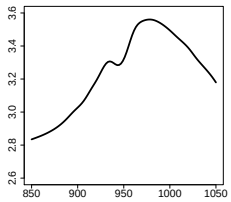
Variable fonctionnelle $\mathcal{X}$	Réponse réelle Y
Spectre infrarouge $\mathcal{X}_1$ du premier morceau de viande 	Y_1 = Taux de matières grasses du premier morceau de viande = 22,5
Spectre infrarouge $\mathcal{X}_2$ du deuxième morceau de viande 	Y_2 = Taux de matières grasses du deuxième morceau de viande = 40,1
⋮	⋮

FIGURE 1.7 – Appariement des couples du jeu de données *Pork*.

spectre $\mathcal{X}_i$ des données *Pork* est associée une variable Y_i (obtenue par un procédé chimique) indiquant le taux de matières grasses contenue dans le morceau de viande correspondant. La figure 1.7 illustre l'appariement de chaque taux de matières grasses à chaque spectre. De façon analogue, les données *Gasoline* sont composées de couples associant à chaque spectre une variable réelle indiquant l'indice d'octane correspondant à l'échantillon de carburant, comme l'illustre la figure 1.8. Pour ces deux exemples d'un point de vue pratique, obtenir un spectre dans le proche infrarouge est bien moins coûteux que le procédé chimique permettant la mesure de la réponse Y (taux de matières grasses ou indice d'octane). D'où l'idée de déterminer la réponse Y à partir du spectre $\mathcal{X}$ en étudiant la relation entre $\mathcal{X}$ et Y. Ceci correspond à l'objectif de la régression fonctionnelle : on cherche à prédire la variable Y_{n+1} associée à une nouvelle réalisation fonctionnelle $\mathcal{X}_{n+1}$. Autrement dit, on cherche à estimer Y_{n+1} par $\hat{Y}_{n+1} = \hat{r}(\mathcal{X}_{n+1})$, où $\hat{r}$ est un estimateur de l'opérateur de régression fonctionnelle r. La modélisation de cet opérateur de régression dépend de la nature du modèle. De façon générale, les modèles de régression fonctionnelle peuvent être classés en trois catégories : le modèle fonctionnel linéaire, les modèles fonctionnels semi-paramétriques et les modèles fonctionnels non-paramétriques.

Dans le cadre du modèle fonctionnel de régression linéaire, l'opérateur de régression r s'écrit sous la forme $r(\mathcal{X}) = \int \theta(t)\, \mathcal{X}(t)\, \mathrm{d}t$, où $\theta \in F$ est un paramètre fonctionnel inconnu à estimer. Le modèle fonctionnel linéaire a été largement étudié dans la littérature (voir par exemple [34, 28, 36, 103, 177, 53, 118, 151]) et appliqué à divers domaines (énergie [97], météorologie [118, 53], entre autres). L'avantage d'un tel modèle est de proposer une interprétation des résultats à travers l'estimation du paramètre fonctionnel θ (et de sa représentation). En revanche, le modèle de régression linéaire fonctionnel souffre parfois d'un manque de flexibilité à cause de la contrainte de linéarité (notamment lorsque le lien entre le prédicteur fonctionnel $\mathcal{X}$ et la réponse Y n'est pas de nature linéaire).

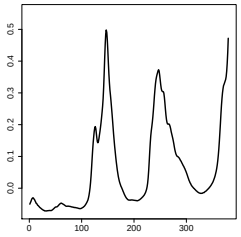
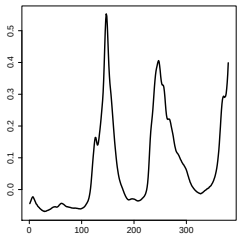
Variable fonctionnelle $\mathcal{X}$	Réponse réelle Y
Spectre infrarouge $\mathcal{X}_1$ du premier échantillon de carburant 	Y_1 = Indice d'octane du premier échantillon de carburant = 85,3
Spectre infrarouge $\mathcal{X}_2$ du deuxième échantillon de carburant 	Y_2 = Indice d'octane du deuxième échantillon de carburant = 85,25
$\vdots$	$\vdots$

FIGURE 1.8 – Appariement des couples du jeu de données *Gasoline*.

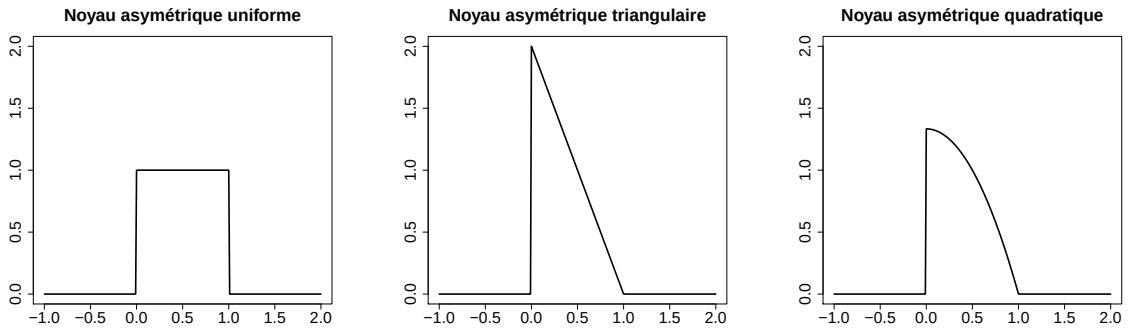
C'est pourquoi la communauté statistique s'est efforcée de développer des modèles de régression beaucoup plus flexibles en s'affranchissant de la contrainte de linéarité et en utilisant des hypothèses de modèles les plus générales possibles. Ces modèles sont appelés modèles de régression non-paramétriques fonctionnels [87]. L'opérateur de régression r, espérance conditionnelle de Y sachant $\mathcal{X}$, est estimé par une extension fonctionnelle de l'estimateur de régression à noyau de Nadaraya-Watson [160, 213] :

$$\hat{r}(\mathcal{X}) = \frac{\sum_{i=1}^{n} Y_i \; K(h^{-1} \; \delta(\mathcal{X}, \mathcal{X}_i))}{\sum_{i=1}^{n} K(h^{-1} \; \delta(\mathcal{X}, \mathcal{X}_i))},$$

où K est un noyau asymétrique, $h \in \mathbb{R}_+^*$ est un paramètre de lissage et δ définit une mesure de proximité entre deux objets fonctionnels. Dans cet estimateur, le noyau asymétrique K a un rôle de pondération. En effet, une réécriture de cet estimateur nous le révèle :

$$\hat{r}(\mathcal{X}) = \sum_{i=1}^{n} w_i(\mathcal{X}) \; Y_i, \text{ avec } w_i(\mathcal{X}) = \frac{K(h^{-1} \; \delta(\mathcal{X}, \mathcal{X}_i))}{\sum_{j=1}^{n} K(h^{-1} \; \delta(\mathcal{X}, \mathcal{X}_j))}.$$

Ce noyau va attribuer pour le calcul de l'estimateur un poids w_i d'autant plus important que la courbe $\mathcal{X}_i$ est proche de la courbe $\mathcal{X}$ en laquelle on évalue l'opérateur de régression. Par ailleurs, il est facile de voir que pour une fonction noyau K de support $[0\,;1]$, le calcul de $\hat{r}(\mathcal{X})$ n'utilisera que les courbes $\mathcal{X}_i$ telles que $\delta(\mathcal{X}, \mathcal{X}_i) \leq h$. Ceci nous permet de comprendre le rôle local joué par le paramètre de lissage h : plus h est grand, et plus le nombre de courbes voisines de $\mathcal{X}$ impliquées dans l'estimation de r sera grand. La figure 1.9 montre quelques exemples de noyaux asymétriques définis sur $[0\,;1]$: uniforme (a), triangulaire (b) et quadratique (c). Tout comme ces trois exemples, il est possible de construire des noyaux asymétriques à partir de noyaux



(a) (b) (c)

FIGURE 1.9 – Quelques exemples de noyaux asymétriques : uniforme (a), triangulaire (b) et quadratique (c).

symétriques, en annulant la partie de support négatif et en doublant la partie de support positif. Le choix d'un noyau symétrique pour cet estimateur aurait été inapproprié ; en effet, l'argument auquel il s'applique dans l'estimateur étant toujours positif, il est tout à fait logique d'adapter le support du noyau en conséquence. Il a cependant été démontré d'un point de vue théorique que l'influence du noyau K sur la vitesse de convergence de cet estimateur est négligeable devant celle des deux autres paramètres h et δ [87]. Dans la pratique, le choix du noyau est souvent arbitraire. En revanche, le paramètre de lissage h joue un rôle essentiel en régulant la prise en compte de courbes plus ou moins proches de celle en laquelle l'estimateur est évalué. La mesure de proximité fonctionnelle δ mérite une attention particulière. En effet, de nombreuses applications de modèles non-paramétriques fonctionnels montrent que leurs performances dépendent fortement du choix de cette mesure δ selon la nature des données étudiées [203, 186]. La régression non-paramétrique fonctionnelle a été appliquée à divers domaines dont la spectrométrie [83, 25] ou la reconnaissance d'écriture [93], entre autres. La très grande flexibilité de ces modèles induit cependant un manque d'interprétabilité.

Pour pallier ce manque tout en conservant une flexibilité suffisamment importante, une autre famille de modèles appelés modèles fonctionnels semi-paramétriques a été introduite. Ces modèles permettent de prendre en compte des relations non-linéaires tout en introduisant des paramètres fonctionnels interprétables. C'est le cas notamment des modèles fonctionnels à directions révélatrices [43, 80] où l'opérateur r se décompose de la façon suivante : $r(\mathcal{X}) = \mu + g_1(\int \theta_1(t)\ \mathcal{X}(t)\ \mathrm{d}t) + \ldots + g_p(\int \theta_p(t)\ \mathcal{X}(t)\ \mathrm{d}t)$ où $g_1, \ldots, g_p$ sont des fonctions inconnues de $\mathbb{R}$ dans $\mathbb{R}$ et $\theta_1, \ldots, \theta_p$ sont des paramètres fonctionnels également inconnus. Ces modèles permettent une plus grande flexibilité et un meilleur potentiel de prédiction que les modèles fonctionnels purement linéaires mais au prix d'une plus grande complexité. Ces modèles opèrent ainsi une décomposition de la variable réponse Y dans divers sous-espaces non-linéairement liés à certaines composantes de la variable fonctionnelle $\mathcal{X}$. Un cas particulier de ces modèles à directions révélatrices est le modèle fonctionnel à indice simple [4, 2, 43] pour lequel $p = 1$: $\hat{r}(\mathcal{X}) = \mu + g(\int \theta(t)\ \mathcal{X}(t)\ \mathrm{d}t)$.

Pour conclure cette section, notons que dans cette thèse, on s'intéressera plus spécifiquement aux modèles de régression non-paramétrique fonctionnelle.

1.1.3 Problèmes de classification dans un cadre fonctionnel

Nous choisissons de nous placer dans un cadre de résolution de problèmes de classification supervisée. On cherche à séparer les données $\mathcal{X}_1, \ldots, \mathcal{X}_n$ en C groupes distincts, autrement dit,

on cherche à prédire une variable catégorielle $L \in \{1, \ldots, C\}$ à partir d'une variable fonctionnelle $\mathcal{X}$. De façon analogue à la régression, on dispose de n couples $(\mathcal{X}_i, L_i)_{1 \leq i \leq n}$ indépendants et identiquement distribués selon une loi inconnue du couple de variables aléatoires $(\mathcal{X}, L)$, où $\mathcal{X}_i$ (respectivement L_i) est la i-ème réalisation de la variable fonctionnelle $\mathcal{X}$ (respectivement de la variable catégorielle ou classe d'appartenance L). Prenons en exemple les données *Canadian Weather*, pour lesquelles à chaque station (à chaque courbe) est associée l'une des quatre régions climatiques du Canada (Atlantique, Pacifique, Continental, Arctique). Dans ce cas précis, il est possible de classifier les 35 stations en 4 groupes uniquement à partir des courbes de températures ou de précipitations. À partir d'une nouvelle courbe $\mathcal{X}_{n+1}$, on cherche à estimer les C probabilités conditionnelles $p_1(\mathcal{X}_{n+1}) := \mathbb{P}(L = 1|\mathcal{X}_{n+1}), \ldots, p_C(\mathcal{X}_{n+1}) := \mathbb{P}(L = C|\mathcal{X}_{n+1})$ d'appartenance à chacune des classes $1, \ldots, C$, pour ensuite prédire la classe L_{n+1} correspondant à celle ayant obtenu la plus grande probabilité d'appartenance estimée $\hat{p}_c(\mathcal{X}_{n+1})$ (règle du maximum a posteriori) : $\hat{L}_{n+1} = \underset{c \in \{1,\ldots,C\}}{\arg\max}\, \hat{p}_c(\mathcal{X}_{n+1})$. Chaque probabilité conditionnelle $p_c(\mathcal{X}_{n+1})$ pouvant s'écrire sous la forme d'une espérance conditionnelle $\mathbb{E}(Y|\mathcal{X}_{n+1})$ avec $Y = \mathbf{1}_{L=c}$ ($Y = 1$ si $L = c$ et $Y = 0$ sinon), résoudre un problème de classification supervisée revient en réalité à résoudre plusieurs (ici C) problèmes de régression à travers l'estimation de ces C espérances conditionnelles. Ainsi, dans le cadre du modèle non-paramétrique fonctionnel, on adapte l'estimateur de Nadaraya-Watson en estimant $p_1, \ldots, p_C$ par :

$$\forall c \in \{1, \ldots, C\},\ \hat{p}_c(\mathcal{X}) = \frac{\sum_{i=1}^{n} \mathbf{1}_{[L_i=c]}\ K(h^{-1}\ \delta(\mathcal{X}, \mathcal{X}_i))}{\sum_{i=1}^{n} K(h^{-1}\ \delta(\mathcal{X}, \mathcal{X}_i))}.$$

De façon analogue, on pourrait ainsi adapter d'autres modèles de régression à la classification supervisée. La classification supervisée de données fonctionnelles a notamment été étudiée dans de nombreux domaines, comme par exemple en entomologie [159], en biologie [136], en médecine [54, 6], dans le domaine de la spectrométrie [179] ou celui de la reconnaissance vocale [105, 87, 89].

1.1.4 Notion de proximité pour des objets fonctionnels

L'une des difficultés de l'approche statistique fonctionnelle vient de la nécessité de définir une «distance» (i.e., mesure de proximité) entre deux objets fonctionnels. En effet, si toutes les distances sont équivalentes dans un espace de dimension finie, cette propriété n'est plus valable dans un espace fonctionnel (i.e., dans un espace de dimension infinie). Ainsi, l'utilisation d'une distance standard peut provoquer l'apparition du «fléau de la dimension». Par exemple, la distance fonctionnelle standard L^2, définie par $d^{L^2}(\mathcal{X}_{i_1}, \mathcal{X}_{i_2}) = \sqrt{\int \left(\mathcal{X}_{i_1}(t) - \mathcal{X}_{i_2}(t)\right)^2 \mathrm{d}t}$, n'échappe pas à cette règle. L'utilisation d'une pseudométrique, très souvent désignée dans la littérature par le terme «semimétrique», au lieu d'une distance standard permet ainsi d'éviter ce phénomène. En considérant un espace fonctionnel F, une pseudométrique δ est une application définie sur $F \times F$ et à valeurs dans $\mathbb{R}_+$ telle que :

- $\forall \mathcal{X}_i \in F,\ \delta(\mathcal{X}_i, \mathcal{X}_i) = 0$,
- $\forall\, (\mathcal{X}_{i_1}, \mathcal{X}_{i_2}) \in F \times F,\ \delta(\mathcal{X}_{i_1}, \mathcal{X}_{i_2}) = \delta(\mathcal{X}_{i_2}, \mathcal{X}_{i_1})$,
- $\forall\, (\mathcal{X}_{i_1}, \mathcal{X}_{i_2}, \mathcal{X}_{i_3}) \in F \times F \times F,\ \delta(\mathcal{X}_{i_1}, \mathcal{X}_{i_2}) \leq \delta(\mathcal{X}_{i_1}, \mathcal{X}_{i_3}) + \delta(\mathcal{X}_{i_3}, \mathcal{X}_{i_2})$.

Ainsi, une distance peut être vue comme une pseudométrique qui vérifie également l'axiome : $\forall\, (\mathcal{X}_{i_1}, \mathcal{X}_{i_2}) \in F \times F, d(\mathcal{X}_{i_1}, \mathcal{X}_{i_2}) = 0 \Rightarrow \mathcal{X}_{i_1} = \mathcal{X}_{i_2}$. L'utilisation d'une pseudométrique au lieu d'une distance revient alors à réduire la dimension de l'espace fonctionnel en autorisant à considérer comme identiques certains objets fonctionnels en réalité différents, limitant ainsi l'impact du «fléau de la dimension».

Les trois principales familles de pseudométriques les plus utilisées dans la littérature statistique fonctionnelle sont respectivement basées sur la dérivation, l'analyse en composantes

principales [122] et les moindres carrés partiels [212]. À titre d'exemple motivant l'introduction de cette première famille de pseudométriques, de nombreuses études présentant des applications sur des données spectrométriques (comme les données *Pork* par exemple) ont constaté que de bien meilleurs résultats pouvaient être obtenus en considérant comme variable explicative fonctionnelle la dérivée seconde des courbes brutes [188, 7, 150, 219]. La famille de pseudométriques δ^{deriv} basée sur la dérivation se définit ainsi :

$$\delta_m^{deriv}(\mathcal{X}_{i_1}, \mathcal{X}_{i_2}) = \sqrt{\int \left(\mathcal{X}_{i_1}^{(m)}(t) - \mathcal{X}_{i_2}^{(m)}(t)\right)^2 \, \mathrm{d}t},$$

où $\mathcal{X}_i^{(m)}$ désigne la dérivée d'ordre m de la fonction $\mathcal{X}_i$. Ainsi, cette famille de pseudométriques est indexée par un entier m. Nous pouvons remarquer que la distance L^2 est un cas particulier de cette famille lorsque $m = 0$. Le calcul répété des dérivées successives d'une fonction étant numériquement très sensible, il est préférable d'opérer une décomposition préalable des fonctions $\mathcal{X}_{i_1}$ et $\mathcal{X}_{i_2}$ dans une base de fonctions dont on connaît les dérivées exactes, comme par exemple la base B-spline [14, 182]. La famille de pseudométriques δ^{fpca} basée sur l'analyse en composantes principales fonctionnelle et indexée par un entier q indiquant le nombre de composantes retenues s'écrit :

$$\delta_q^{fpca}(\mathcal{X}_{i_1}, \mathcal{X}_{i_2}) = \sqrt{\sum_{k=1}^{q} \left(\int \left[\mathcal{X}_{i_1}(t) - \mathcal{X}_{i_2}(t)\right] u_k(t) \, \mathrm{d}t\right)^2},$$

où $u_1, \ldots, u_q$ sont déduits de l'analyse en composantes principales de la variable fonctionnelle dont les $\mathcal{X}_i$ sont les réalisations. L'analyse en composantes principales fonctionnelle consiste à décomposer la variable fonctionnelle $\mathcal{X}$ selon des directions orthonormées $u_1, u_2, \ldots$ telles que $Var\left(\int u_1(t)\, \mathcal{X}(t)\, \mathrm{d}t\right)$ est maximale, $Var\left(\int u_2(t)\, \mathcal{X}(t)\, \mathrm{d}t\right)$ est maximale avec $\int u_1(t)\, u_2(t)\, \mathrm{d}t = 0$, $\ldots$, $Var\left(\int u_q(t)\, \mathcal{X}(t)\, \mathrm{d}t\right)$ est maximale avec pour $j = 1, \ldots, q-1$, $\int u_j(t)\, u_q(t)\, \mathrm{d}t = 0$. L'analyse en composantes principales maximise la part de variance expliquée par le sous-espace vectoriel engendré par $u_1, \ldots, u_q$. Cependant, la construction de δ^{fpca} ne dépend pas de la variable réponse Y. Or, dans un contexte où le centre d'intérêt est le lien qui relie la variable réponse Y à la variable fonctionnelle $\mathcal{X}$, il est raisonnable de penser que cette famille de pseudométriques puisse ne pas être la plus adaptée à la construction de l'estimateur non-paramétrique fonctionnel. C'est pourquoi une troisième famille de pseudométriques δ^{mplsr}, basée sur une décomposition des moindres carrés partiels, a été introduite. Cette famille de pseudométriques est définie par :

$$\delta_q^{mplsr}(\mathcal{X}_{i_1}, \mathcal{X}_{i_2}) = \sqrt{\sum_{k=1}^{q} \left(\int \left[\mathcal{X}_{i_1}(t) - \mathcal{X}_{i_2}(t)\right] v_k(t) \, \mathrm{d}t\right)^2},$$

où $v_1, \ldots, v_q$ sont déduits de la décomposition des moindres carrés partiels (dépendant de $\mathcal{X}$ et Y) et l'entier q indexant cette famille indique le nombre de composantes retenues. La décomposition des moindres carrés partiels peut être vue comme une extension de l'analyse en composantes principales prenant également en compte la covariabilité entre $\mathcal{X}$ et Y, au lieu de la seule variabilité de $\mathcal{X}$. Ainsi, la variable fonctionnelle $\mathcal{X}$ est décomposée selon des directions orthonormées $v_1, v_2, \ldots$ telles que $Cov^2\left(Y, \int v_1(t)\, \mathcal{X}(t)\, \mathrm{d}t\right)$ est maximale, $Cov^2\left(Y, \int v_2(t)\, \mathcal{X}(t)\, \mathrm{d}t\right)$ est maximale avec $\int v_1(t)\, v_2(t)\, \mathrm{d}t = 0$, $\ldots$, $Cov^2\left(Y, \int v_q(t)\, \mathcal{X}(t)\, \mathrm{d}t\right)$ est maximale avec pour $j = 1, \ldots, q-1$, $\int v_j(t)\, v_q(t)\, \mathrm{d}t = 0$. L'algorithme des moindres carrés partiels cherche ainsi à maximiser à la fois la variance de $\mathcal{X}$ et les corrélations entre $\mathcal{X}$ et Y.

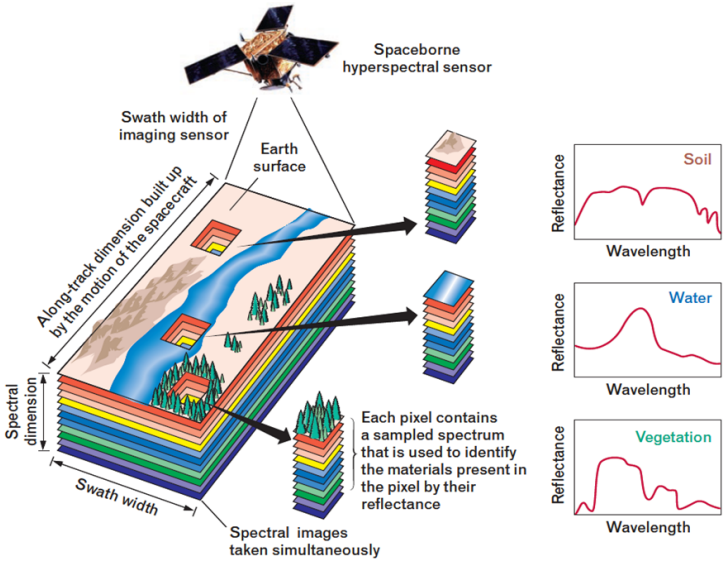


FIGURE 1.10 – Principe de fonctionnement des capteurs hyperspectraux [187].

1.2 Image de télédétection hyperspectrale

La télédétection est l'ensemble des techniques de mesures distantes et sans contact sur des objets dont on souhaite connaître des caractéristiques physiques et/ou biologiques. Les données recueillies par ces mesures peuvent être représentées sous forme d'images numériques. De façon générale, une image de télédétection se décompose selon trois axes d'échantillonnage : l'échantillonnage spectral, l'échantillonnage spatial et l'échantillonnage temporel. L'échantillonnage spectral concerne la décomposition d'une ou plusieurs partie(s) du spectre électromagnétique dans les domaines du visible, du proche infrarouge et du moyen infrarouge. L'échantillonnage spatial est défini par la taille de chaque pixel de l'image. L'échantillonnage temporel se caractérise par la durée entre deux acquisitions successives d'images de la même zone. Une image couleur peut être vue comme un exemple d'image de télédétection : elle est composée de 3 bandes chromatiques (le rouge, le vert et le bleu). À chaque pixel de cette image correspond un triplet de valeurs (r, g, b). Il s'agit là d'un exemple d'image multispectrale à 3 canaux spectraux. De façon plus générale, une image multispectrale est produite par des capteurs mesurant l'énergie réfléchie par un environnement donné pour quelques bandes spécifiques du spectre électromagnétique (le plus souvent entre 3 et 10).

Les progrès techniques en matière de capteurs de télédétection permettent d'obtenir des résolutions de plus en plus fines suivant chacun des trois axes d'échantillonnage. Par exemple, une image hypertemporelle est caractérisée par un échantillonnage très fin dans le domaine temporel, c'est-à-dire à une importante succession d'images prises dans un laps de temps relativement court. De même, une image à très haute résolution spatiale fait référence à une image dont la taille des pixels est typiquement inférieure au mètre carré. De façon analogue, une image hyperspectrale se caractérise par la finesse de son échantillonnage dans le domaine spectral [189].

Ainsi, chaque image de télédétection se caractérise par plusieurs propriétés associées à chacun des trois axes :

- La couverture spectrale (intervalles du domaine spectral balayés par le capteur),
- Le nombre de bandes spectrales,
- La largeur de chaque bande spectrale,

Type de capteur	**Nom du capteur**	**Nombre de bandes spectrales**	**Couverture spectrale (en nm)**	**Résolution spectrale (en nm)**	**Résolution spatiale (en m)**
Capteurs aéroportés	AVIRIS	224	400-2450	10	4/20
	CASI	72	400-944	76	plus de 2,5
	ROSIS	115	430-860	4	plus de 1,7
	HYSPEC	160	400-2450	4,5	0,5
Capteurs satellitaires	HYPERION	242	400-2500	10	30
	HySI	64	400-950	10	506
	HJ-1A	110-128	450-950	5	100
	CHRIS	37-118	415-1050	2/10	25/50

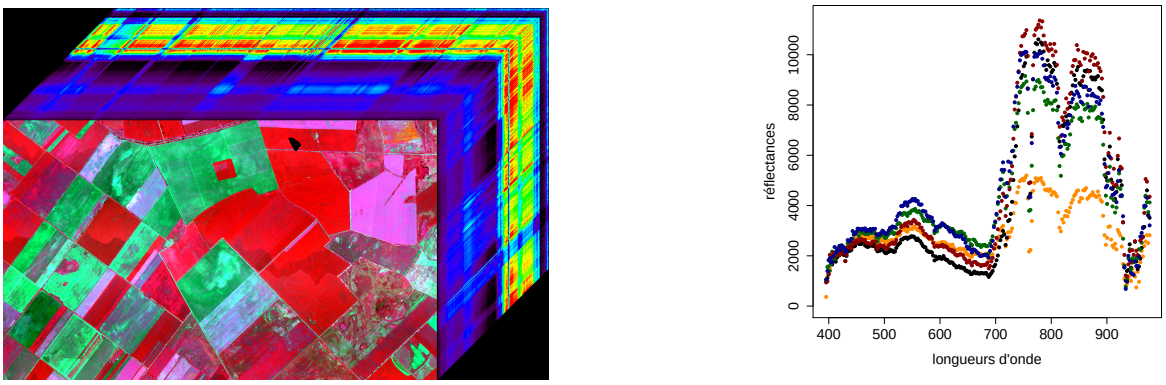
TABLE 1.1 – Caractéristiques spectrales et spatiales de divers capteurs hyperspectraux aéroportés et satellitaires.

- La résolution spatiale (taille réelle correspondant à un pixel de l'image),
- La résolution temporelle (fréquence de relevés des données par le capteur).

En télédétection, il existe essentiellement deux catégories de capteurs : les capteurs actifs et les capteurs passifs. La principale différence existant entre ces deux catégories provient de la nature du signal lumineux capté. Les capteurs actifs émettent leurs propres signaux selon différents canaux spectraux et récupèrent les signaux réfléchis. Les capteurs passifs ne sont que des récepteurs : ils captent la lumière du soleil réfléchie par l'environnement pour la décomposer selon différents canaux spectraux. La figure 1.10 illustre le principe de fonctionnement des capteurs hyperspectraux. Balayant la zone d'études par bandes spatiales d'une largeur donnée (dépendante du capteur), les capteurs hyperspectraux effectuent des mesures de réflectances en continu sur une partie du spectre électromagnétique, permettant ainsi de capter de subtiles variations de l'énergie réfléchie. Ces capteurs opèrent une acquisition simultanée de centaines, voire de milliers d'images spectrales monochromes (une image par bande spectrale). Ainsi, chaque pixel est décomposé selon un échantillonage spectral de réflectances permettant de connaître la nature de ce qu'il contient. Deux types de capteurs sont principalement utilisés en imagerie hyperspectrale : les capteurs aéroportés (AVIRIS, CASI, ROSIS, HYSPEX,...) et les capteurs satellitaires (HYPERION, HySI, HJ-1A, CHRIS, ...). La table 1.1 présente les caractéristiques spectrales et spatiales de chacun de ces capteurs. Partageant des caractéristiques spectrales similaires (nombre de bandes spectrales, couverture spectrale, résolution spectrale), la principale différence entre ces deux types de capteurs provient de leur résolution spatiale (taille réelle correspondant à la longueur d'un côté du pixel de l'image), causée par la différence d'altitude à laquelle ils sont placés. Ainsi, tandis que les capteurs satellitaires ont une résolution spatiale supérieure à 30 mètres, celle des capteurs aéroportés est bien inférieure et peut descendre (selon les capteurs) jusqu'à 50 centimètres.

Une image hyperspectrale peut se représenter en 3 dimensions sous la forme d'un cube de données. Les deux premières dimensions sont associées à la position spatiale de chaque pixel de l'image (i.e., latitude et longitude). Quant à la troisième dimension, elle représente la dimension spectrale des images (en analogie avec les 3 couleurs de la décomposition RGB). La figure 1.11 illustre cette représentation : à chaque pixel de l'image (a) est associé un vecteur contenant les valeurs relevées pour chaque bande spectrale (b). On obtient ainsi des spectres de réflectance représentant l'intensité de lumière réfléchie par la zone couverte par chaque pixel pour chaque longueur d'onde considérée.

L'imagerie hyperspectrale est utilisée pour couvrir divers types de zones terrestres : les océans



(a) (b)

FIGURE 1.11 – Représentation d'une image hyperspectrale (a) et données spectrales extraites pour quelques pixels (b) (une couleur différente pour chaque pixel).

[92, 55], le milieu urbain [178, 60], les sols [8, 132, 98] ou la végétation [180, 199]. Ces images trouvent des applications dans de nombreux domaines comme par exemple l'écologie [52, 167, 95], le domaine militaire [148], l'agriculture [13, 134] ou l'hydrologie [181].

L'intérêt de l'utilisation de l'imagerie hyperspectrale réside dans la grande quantité et la finesse de l'information spectrale qu'elle contient. L'étude d'images hyperspectrales conduit au traitement d'un important volume de données de grande dimension spectrale car chaque pixel est représenté par un vecteur comportant autant de variables que de bandes spectrales. Seulement, l'acquisition de données hyperspectrales est un processus coûteux, tant en termes de temps et de moyens humains (organisation d'une campagne de terrain pendant une ou plusieurs semaines) qu'en termes purement financiers (coût de 80000 euros pour 250 km^2). Dans une optique d'étude statistique de telles données, cette contrainte conduit au développement de modèles contruits sur un échantillon restreint de pixels. Cependant, cette configuration (nombre de bandes spectrales grand par rapport à la taille de l'échantillon d'apprentissage) correspond à celle du «fléau de la dimension», provoquant une dégradation de la qualité de prédiction des méthodes multivariées.

L'étude d'images hyperspectrales a conduit à la résolution de problèmes divers et variés tels que la régression, la classification, la détection, la segmentation ou le démélange. Les principes de la régression et de la classification supervisée, visant repectivement à prédire une variable scalaire ou catégorielle, ont déjà été abordés dans les sections 1.1.2 et 1.1.3 dans le cadre de l'étude de données fonctionnelles. La détection d'une ou plusieurs cibles dans une image vise au repérage et à l'identification d'objets spécifiques. La détection automatique est un réel défi car son algorithme doit s'adapter à toutes sortes de contraintes telles que les changements de point de vue, de taille, d'échelle ou l'obstruction partielle. La détection a été appliquée dans divers contextes en imagerie hyperspectrale [149, 191, 148], comme par exemple dans le domaine militaire pour la détection de cibles. La segmentation est un procédé cherchant à séparer une image en plusieurs groupes de pixels proches, partageant au sein de chaque groupe des caractéristiques communes. Ainsi, cette approche se base sur la composante spatiale des images et peut être combinée avec des modèles construits sur la dimension spectrale. Il existe plusieurs types de segmentation, respectivement basés sur les régions, les contours ou le seuillage de l'intensité des pixels de l'image. La segmentation d'images hyperspectrales a été étudiée au travers de nombreuses applications [196, 139]. Les problèmes de démélange spectral visent à déterminer

la composition (en termes de proportions) de chaque pixel contenant des mélanges de différents matériaux juxtaposés et/ou intimement mélangés. Ces modèles sont basés sur l'hypothèse selon laquelle chaque hyperspectre observé est une combinaison d'hyperspectres purs souvent inconnus. Les problèmes de démélange dans un cadre hyperspectral ont largement été étudiés depuis une dizaine d'années [101, 12]. De nombreux travaux font état d'études d'images hyperspectrales aéroportées couplant la dimension spectrale avec la dimension spatiale par la modélisation de dépendances entre pixels proches [74, 194, 195, 140]. Dans cette thèse, nous avons mis en œuvre le traitement statistique de données hyperspectrales en se consacrant à la dimension spectrale de ces données.

Diverses méthodes multivariées ont été développées afin de résoudre les problèmes statistiques de régression et de classification supervisée :

- Les méthodes de représentation à noyau [29] (comme les Séparateurs à Vaste Marge [209] par exemple) sont des méthodes opérant une transformation de l'espace de départ dans un espace de dimension plus grande. En pratique, il suffit d'effectuer le calcul des produits scalaires entre deux spectres à l'aide d'une fonction noyau K (cette astuce est appelée «Kernel trick»), au lieu de les projeter et de calculer les produits scalaires dans le nouvel espace. Ainsi, un problème dont le lien statistique entre la variable à prédire et les variables explicatives n'est pas linéaire dans l'espace de départ peut devenir linéaire dans ce nouvel espace ; le choix de la fonction noyau est donc déterminant. Quelques exemples d'application de méthodes à noyau sur des données hyperspectrales sont disponibles dans [29, 131, 100, 45].
- Une autre approche consiste à effectuer une réduction préalable de la dimension des données en opérant une diminution du nombre de variables. Cette approche est divisée en deux catégories : les méthodes de sélection de variables et les méthodes d'extraction de caractéristiques. Les méthodes de sélection de variables cherchent à constituer le meilleur sous-ensemble des variables de départ $\{X(\lambda_1), \ldots, X(\lambda_d)\}$ pour la construction d'un modèle conservant une bonne capacité de prédiction. Cette approche réduit mais ne supprime cependant pas les fortes corrélations entre les variables, omniprésentes lors de l'étude de données hyperspectrales. Les méthodes d'extraction de caractéristiques (comme par exemple l'Analyse en Composantes Principales [122]) transforment l'espace de départ de grande dimension en un espace de dimension réduite de façon à obtenir de nouvelles variables décorrélées dans ce nouvel espace à partir des variables de départ. Ainsi, un tel prétraitement des données permet l'application de méthodes multivariées standard sur ces nouvelles variables. [114, 65, 1] présentent quelques exemples de méthodes de réduction de la dimension appliquées à l'étude de données hyperspectrales. Il est possible de combiner ce type de méthodes avec le «Kernel trick» afin d'opérer une réduction non-linéaire de la dimension [75, 130].
- Un autre type de méthodes permettant le traitement de données en grande dimension est basé sur une régularisation des méthodes, c'est-à-dire l'introduction de contraintes supplémentaires lors de l'apprentissage du modèle, permettant ainsi de résoudre un problème mal conditionné ou d'éviter le surapprentissage. Dans un cas standard, cette approche consiste à altérer un problème d'optimisation initial par l'ajout d'un terme de régularisation dont l'influence est contrôlée par un paramètre réel. Des méthodes de régularisation appliquées à des données hyperspectrales sont présentées dans [146, 218]. Ces méthodes peuvent également être combinées avec le «Kernel trick» [127].

Étant données les carastéristiques des données hyperspectrales (nature continue des spectres, fine discrétisation, fortes corrélations des bandes spectrales adjacentes, ...), il est tout à fait légitime de considérer les données hyperspectrales comme un exemple de données fonctionnelles.

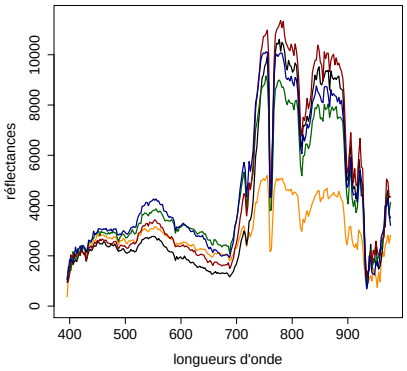


FIGURE 1.12 – Reconstruction continue d'hyperspectres à partir des données discrétisées.

Ainsi, l'étude statistique de données hyperspectrales par des méthodes fonctionnelles devient une alternative intéressante à l'application de méthodes multivariées. Seulement, les méthodes fonctionnelles n'ont été que très récemment investiguées pour la résolution de problèmes impliquant des données hyperspectrales [124, 193, 161, 61, 137]. C'est pourquoi nous proposons dans cette thèse le développement de méthodes fonctionnelles et leur application à l'étude statistique de données hyperspectrales.

1.3 Approche fonctionnelle des données hyperspectrales et contributions de la thèse

Étant donnée la nature continue des spectres de réflectance, il est naturel de considérer les données hyperspectrales comme un exemple de données fonctionnelles. La figure 1.12 illustre ce principe en présentant une reconstruction continue des données discrétisées de la figure 1.11 (b). Ainsi, aborder l'analyse d'images hyperspectrales par l'étude de méthodes fonctionnelles pourrait permettre de mieux appréhender cette nature qui n'est pas prise en compte par la plupart des autres méthodes. Les données spectrométriques partagent avec les données hyperspectrales de nombreux points communs caractéristiques des données fonctionnelles (nature continue des spectres, fine discrétisation, fortes corrélations des bandes spectrales adjacentes, ...). Si on trouve dans la littérature des exemples d'application de méthodes fonctionnelles à des données spectrométriques [85, 86, 87, 81, 79], ces méthodes n'ont été que très récemment utilisées pour l'étude de données hyperspectrales. Nous proposons dans cette thèse l'étude systématique de données hyperspectrales à travers le prisme de méthodes statistiques modernes, avec une attention particulière pour les méthodes non-paramétriques fonctionnelles.

Le travail effectué au cours de cette thèse a mis en évidence l'intérêt de l'utilisation de méthodes non-paramétriques fonctionnelles pour l'étude statistique d'images hyperspectrales en proposant une approche complémentaire à celle induite par les méthodes multivariées standard. Ces méthodes permettent notamment l'exploitation complète de la nature fonctionnelle des données (continuité, ordre des bandes spectrales, fortes corrélations, ...), ce que ne permettent pas les méthodes multivariées.

Le chapitre 2 de ce manuscrit s'intéresse à la classification supervisée de données hyperspectrales au moyen de méthodes fonctionnelles et non-fonctionnelles. Diverses techniques mul-

tivariées telles que le modèle multinomial logistique, les Modèles de Mélanges Gaussiens, les Séparateurs à Vaste Marge [209] et les Forêts aléatoires [21] ont été comparées avec une méthode non-paramétrique fonctionnelle sur plusieurs jeux de données hyperspectrales. Pour le calcul de l'estimateur non-paramétrique fonctionnel, l'utilisation de pseudométriques au lieu de distances usuelles a été abordée. Cette approche permet d'obtenir une alternative intéressante afin de réduire l'impact du «fléau de la dimension», d'autant plus lorsque la taille de l'échantillon alloué à la construction du modèle est limitée. Dans ce cadre, la pseudométrique basée sur la décomposition des moindres carrés partiels (mplsr) semble particulièrement adaptée à l'étude d'images hyperspectrales. Diverses configurations faisant intervenir l'influence de la taille, de la composition et du niveau de bruit dans les labels ont permis de conclure que la qualité de prédiction du modèle dépend également de la nature des données et du problème statistique fonctionnel à résoudre. Des variances inter-classes nettement plus fortes que les variances intra-classes et une absence de bruit dans les données profitent aux méthodes multivariées standard, tandis que des classes donc la variabilité interne est comparable à la variance inter-classes, des données bruitées et un échantillon d'apprentissage de petite taille révèlent tout le potentiel de discrimination des méthodes non-paramétriques fonctionnelles.

Le chapitre 3 propose l'étude de données hyperspectrales dans un cadre de modèle prédictif multivarié parcimonieux (régression ou classification supervisée). Il est important de noter qu'ici, contrairement aux autres développements réalisés dans cette thèse, la nature fonctionnelle des données n'a pas été prise en compte. Deux travaux distincts sont présentés dans ce chapitre. Le premier s'intéresse à deux méthodes parcimonieuses, l'une linéaire (LASSO [202]) et l'autre non-paramétrique («Most Predictive Design Points» [82]), permettant de prédire une réponse réelle à partir d'un hyperspectre. Ce travail permet de souligner (pour les données étudiées) la pertinence de réduire la dimension spectrale en sélectionnant les longueurs d'onde les plus prédictives. Le deuxième propose une autre méthode non-linéaire de sélection de variables avec une extraction basée sur un classifieur de mélanges gaussiens [168]. L'implémentation efficace de cette méthode permet de rendre plus rapide la mise à jour des paramètres et l'accès aux sous-modèles. En comparaison avec la méthode des Séparateurs à Vaste Marge (SVM), cette méthode sélectionne plus rapidement un faible nombre de caractéristiques informatives tout en conservant un potentiel de prédiction similaire et en facilitant l'interprétation des bandes spectrales extraites. Cependant, l'instabilité des solutions obtenues, liée aux fortes corrélations présentes dans ce type de données, implique la nécessité de mener d'autres études plus approfondies sur le sujet.

Le chapitre 4 concerne le traitement de données hyperspectrales bruitées. En pratique, l'acquisition de données hyperspectrales s'accompagne de perturbations engendrant un bruit de mesure plus ou moins intense pouvant varier selon les pixels et les bandes spectrales. De façon générale, les sources de ces perturbations sont multiples : bandes d'absorption dans l'atmosphère causant une atténuation plus ou moins forte du signal, bruit numérique du capteur variant selon la longueur d'onde considérée, corrections géométriques induisant une distorsion du signal, ... Ainsi, le seul fait de ne pas prendre en compte la présence éventuelle de bruit dans les données conduit parfois à une dégradation plus ou moins importante de la capacité de prédiction de méthodes pourtant reconnues comme étant adaptées à l'étude de ce type de données. Dans ce chapitre, une procédure d'estimation en deux étapes est proposée, s'appliquant aussi bien dans un cadre de régression que de classification supervisée. La première étape consiste à lisser les données fonctionnelles bruitées de façon non-paramétrique tandis que la seconde se focalise sur l'estimation non-paramétrique de l'opérateur de régression. Il est important de remarquer que ces deux étapes sont emboîtées de sorte que leurs paramètres respectifs optimisent un critère prédictif global. Des résultats théoriques sur le nouvel estimateur ont permis de retrouver la vitesse de convergence que l'on obtiendrait si l'on observait les données non bruitées. Les résultats

obtenus en pratique sont en accord avec leur pendant théorique : cette méthode imbriquée est d'autant moins impactée par l'intensité du bruit dans les données que le nombre de points de discrétisation est grand devant le nombre de pixels, ce qui est souvent le cas lors de traitements de données hyperspectrales. Cette méthode non-paramétrique fonctionnelle avec lissage a été comparée en pratique avec la méthode non-paramétrique fonctionnelle standard sans lissage et avec des méthodes multivariées usuelles en télédétection. Cette méthode a notamment donné les meilleurs résultats sur des données hyperspectrales réelles supposées bruitées mais dont le profil du niveau de bruit est totalement inconnu.

Le chapitre 5 est consacré à la mise en pratique des méthodes présentées dans ce manuscrit, et plus particulièrement au développement de nouvelles méthodes permettant l'étude de données hyperspectrales. Au cours de cette thèse, nous avons eu l'occasion d'implémenter principalement trois approches fonctionnelles à l'aide du logiciel statistique R [170]. La première est une réécriture de la méthode non-paramétrique fonctionnelle implémentée par [87] afin d'en réduire le temps de calcul. Cette nouvelle implémentation permet notamment un traitement plus rapide des données à l'aide d'une matricialisation des calculs, au prix cependant de la nécessité d'une plus grande capacité de stockage mémoire. La deuxième approche implémentée est une extension du modèle multinomial logistique adaptée au traitement de données fonctionnelles. Cette méthode combine un prétraitement décomposant les données fonctionnelles dans une base de fonctions splines [14] et l'application de l'approche multivariée standard du modèle multinomial logistique. La troisième approche concerne l'implémentation de l'estimateur non-paramétrique fonctionnel dans le cadre de l'étude de données bruitées. Cette approche combine un lissage à noyau des données avec l'estimateur non-paramétrique fonctionnel standard. Une mise en pratique est proposée par l'écriture de codes R pour l'application de ces implémentations sur une partie des données hyperspectrales étudiées au cours de cette thèse. L'accent y est mis sur les protocoles expérimentaux utilisés ainsi que les résultats obtenus à chaque étape. Les données étudiées, les codes R utilisés ainsi que les fichiers d'aide associés sont accessibles depuis le site de DYNAFOR : https://dynafor.toulouse.inra.fr/dynafornet/index.php/fre/Collaborateurs/Doctorants/Zullo.

L'annexe A est dédiée à la présentation des principaux jeux de données hyperspectraux utilisés au cours de cette thèse. L'accent est mis sur leurs principales différences, tant du point de vue des caractéristiques de chaque capteur que des milieux observés, représentatives de la diversité des données hyperspectrales.

Communications écrites et orales

Publications dans des journaux internationaux

- M. Fauvel, C. Dechesne, A. Zullo, and F. Ferraty. Fast forward feature selection of hyperspectral images for classification with Gaussian mixture models. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):2824–2831, 2015.
- A. Zullo, M. Fauvel, and F. Ferraty. Comparison of functional and multivariate spectral-based supervised classification methods in hyperspectral image. *Journal of Applied Statistics*, submitted.
- F. Ferraty, A. Zullo, and M. Fauvel. Nonparametric regression on contamined functional predictor with application to hyperspectral data. *Econometrics and Statistics*, submitted.

Actes de conférences

- A. Zullo, M. Fauvel, F. Ferraty, M. Goulard, and P. Vieu. Non-parametric functional

methods for hyperspectral image classification. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 3422–3425, Quebec city, July 13th-18th, 2014.

- M. Fauvel, A. Zullo, and F. Ferraty. Nonlinear parsimonious feature selection for the classification of hyperspectral images. In *6th Workshop on Hyperspectral image and signal processing: evolution in remote sensing (WHISPERS)*, Lausanne, Switzerland, 24-27 June 2014.

Communications orales

- A. Zullo, M. Fauvel, and F. Ferraty. Classification d'images hyperspectrales par des méthodes fonctionnelles non-paramétriques. In *3ème colloque scientifique du Groupe Hyperspectral de la Société Française de Photogrammétrie et de Télédétection*, Porquerolles, 15-16 mai 2014.
- A. Zullo, F. Fauvel, and F. Ferraty. Sélection de variables pour l'imagerie hyperspectrale. In *46e Journées de Statistique, Société Française de Statistique*, Rennes, 2-6 juin 2014.
- A. Zullo, F. Ferraty, and M. Fauvel. Débruitage d'images hyperspectrales avec un modèle de bruit hétéroscédastique : application à l'estimation de variables biophysiques par régression non-paramétrique fonctionnelle. In *4ème colloque scientifique du Groupe Hyperspectral de la Société Française de Photogrammétrie et de Télédétection*, Grenoble, 11-13 mai 2016.

Chapitre 2

Classification de données hyperspectrales par des méthodes fonctionnelles

La classification supervisée d'images hyperspectrales est rendue difficile par le grand nombre de variables spectrales et par le petit nombre d'échantillons de référence pour la construction du modèle [121]. De nombreuses méthodes ont déjà été proposées pour aborder ce problème (méthodes bayésiennes [133], méthodes d'extraction de caractéristiques [133], forêts aléatoires [104], méthodes à noyau [30], réseaux de neurones [175], ...). Dans ce chapitre, nous présentons et développons des méthodes non-paramétriques fonctionnelles, avec une attention particulière portée sur le choix de la «distance» (pseudométrique) entre les courbes de réflectance, nécessaire au calcul de l'estimateur associé.

La première partie de ce chapitre vise à évaluer la pertinence de l'application d'une méthode non-paramétrique fonctionnelle pour la classification supervisée de données hyperspectrales (partie issue de l'article [223]). Le modèle est basé sur une estimation des probabilités conditionnelles d'appartenance à chacune des classes par un estimateur à noyau. La classe prédite est alors celle dont la probabilité estimée est la plus forte. Cette méthode a été comparée pour divers choix de pseudométriques (distance L^2, pseudométriques «fpca» et «mplsr») à plusieurs méthodes multivariées standard de classification. Parmi ces méthodes, on trouve le modèle multinomial logistique, les Modèles de Mélanges Gaussiens (construits à partir d'une matrice de covariance régularisée) et les Séparateurs à Vaste Marge. Les quatre méthodes ont été appliquées sur deux jeux de données hyperspectrales (*MADONNA* et *University of Pavia*) en restreignant la construction du modèle à un faible nombre de pixels. Le choix de la pseudométrique pour le calcul de l'estimateur non-paramétrique fonctionnel a permis de réduire l'impact du «fléau de la dimension». Dans ce cadre, la pseudométrique basée sur la décomposition des moindres carrés partiels (mplsr) a permis d'obtenir des résultats significativement meilleurs que les autres pseudométriques et les autres méthodes.

La deuxième partie présente un comparatif détaillé de diverses méthodes fonctionnelles et non-fonctionnelles pour la classification supervisée de données hyperspectrales (partie issue de l'article [222]). L'accent est mis sur la prise en compte de la seule dimension spectrale des données. Six méthodes provenant de trois communautés statistiques ont été comparées :

- Les modèles de mélanges :
  - Les modèles de mélanges gaussiens (méthodes basées sur l'hypothèse selon laquelle la densité des spectres s'écrit comme une combinaison linéaire convexe de densités de probabilités gaussiennes)

  - L'analyse discriminante pour la grande dimension (modèle à sous-espaces où chaque classe vit dans un sous-espace vectoriel plus petit que l'espace initial)
- Les méthodes non-paramétriques de type «machine learning» :
  - Les séparateurs à vaste marge (classifieur binaire visant à trouver le meilleur hyperplan séparant les données, étendu à la résolution de problèmes multiclasses)
  - Les forêts aléatoires (classifieur basé sur une importante accumulation d'arbres de décision binaire)
- Les méthodes fonctionnelles :
  - Un modèle multinomial logistique fonctionnel (modèle combinant l'approche multivariée standard de la méthode à une décomposition des données fonctionnelles dans une base de fonctions splines)
  - Une méthode de discrimination non-paramétrique fonctionnelle (modèle basé sur l'estimation des probabilités d'appartenance des spectres à chacune des classes à partir d'un estimateur à noyau)

Toutes ces méthodes ont été appliquées à deux jeux de données hyperspectraux selon un protocole expérimental permettant de les comparer selon trois critères : la fiabilité face à une diminution du nombre total de pixels dans l'échantillon d'apprentissage, la stabilité face à un déséquilibre dans le nombre de pixels de chaque classe alloués à la construction du modèle, ainsi que la robustesse face à différents niveaux de bruit dans les données fonctionnelles. Cette étude approfondie a permis de constater que la qualité de prédiction d'un modèle ne dépend pas uniquement du principe selon lequel il a été contruit, mais également de la nature des données et du problème statistique fonctionnel à résoudre. Des variances inter-classes nettement plus fortes que les variances intra-classes et une absence de bruit dans les données profitent aux méthodes multivariées standard, tandis que des classes donc la variabilité interne est comparable à la variance inter-classes, des données bruitées et un échantillon d'apprentissage de petite taille révèlent tout le potentiel de discrimination des méthodes non-paramétriques fonctionnelles. Cette partie présente et met à disposition l'implémentation et l'application à l'aide du logiciel statistique R [170] des deux méthodes fonctionnelles développées (modèle multinomial logistique fonctionnel et implémentation rapide du modèle non-paramétrique fonctionnel).

Non-parametric functional methods for hyperspectral image classification

A. Zullo[1,2], M. Fauvel[1], F. Ferraty[2], M. Goulard[1] and P. Vieu[2]

[1] Laboratoire DYNAFOR - UMR 1201 - INRA & INP Toulouse, France
[2] Institut de Mathématiques de Toulouse - UMR 5219 & Université de Toulouse, France

Abstract. The objective of this article is to assess the relevance of a statistical method for hyperspectral image classification. We focus on the implementation of a functional method whose main objective is to consider each hyperspectrum as a continuous curve in order to predict its associated class. The implemented functional nonparametric discrimination method is a recently developed technique whose performance are greatly dependent on the choice of a "proximity measure". Behavior in practice of this method has been compared with three more standard others on two sets of hyperspectral data with supervised classification for 50 independent sets using a classification error rate criterion. Experimental results show that this method provides an interesting alternative to conventional methods.

Keywords. Curse of dimensionality, hyperspectral image classification, nonparametric functional model, statistical method.

2.1 Introduction

The classification of hyperspectral images has received a lot of attention in the last decade [77]. In a hyperspectral image, a standard observation for some pixel i leads to a d-dimensional random vector $\mathbf{X}_i = (X_i^1, \ldots, X_i^d)$ corresponding to an hyperspectrum sampled at d wavelengths $\lambda^1, \ldots, \lambda^d$ (i.e., $X_i^j = X_i(\lambda^j)$ for $j \in \{1, \ldots, d\}$) and a categorical response Y_i indicating the label of some class membership. Given a learning set $\{(\mathbf{X}_i, Y_i)$ for $i \in \{1, \ldots, n\}\}$, the problem of "classification" consists in assigning each pixel to a class. This supervised classification of hyperspectral image is a nontrivial task ; most of usual discriminating methods are not appropriate to such datasets [121]. The failure of these methods is mainly caused by the combination of various particular settings. Firstly, sampled hyperspectra have a large number d of spectral variables, typically 160 for our first application case. Secondly, we considered the case with a rather small training set size n (i.e., a small number of pixels) linked to sampling constraint that might apply, in our first application case we used 30 observations by category for an overall number of 12 categories. From a statistical point of view, this amounts to consider a dataset containing a large number d of covariates with a small set size n. This unconfortable situation is known under the terminology "curse of dimensionality" [62].
Several works have been done to better handle the very high number of spectral variables, using Bayesian models [133], feature extraction and feature reduction techniques [133, 26], random forest [104], neural networks [175] or kernel methods [30]. Among these methods, Support Vectors Machines (SVM) have shown very good performances in terms of classification accuracy. However, another particular feature linked to hyperspectra has not been investigated so far. It is related to the high correlation between consecutive variables (i.e., X_i^j and X_i^{j+1}). One way to better handle such statistical problem consists to consider the i^{th} sampled hyperspectrum $\mathbf{X}_i$ as the discretized version of the curve $\{\chi_i(\lambda)$; $\lambda \in [\lambda_{\min}, \lambda_{\max}]\}$, where the word "curve" stands for a real quantity $\chi_i(\lambda)$ varying continuously with the wavelength λ. Formalizing the hyperspectra as curves enjoys the advantage to take into account intrinsic feature like the order of the wavelengths or the shape of the hyperspectra profiles. It also reduces the impact of the curse of dimensionality by means of particular proximity measures called pseudometrics and acting on

the curves space. The objective of this paper is to investigate the functional modelization for the classification of hyperspectral images.
In the following, a nonparametric functional model is presented in Section 2.2. Attention is paid to the measure of closeness between two curves. In Section 2.3, experimental results are provided on one real data set. The proposed model is compared in terms of classification accuracies with other state of the art classifiers : Gaussian Mixture Models (GMM) and Support Vectors Machines (SVM).

2.2 Nonparametric functional model

From now on, $\mathcal{X} := \{\mathcal{X}(\lambda);\, \lambda \in [\lambda_{\min}, \lambda_{\max}]\}$ stands for a random curve, a particular case of mathematical objects called "functional variables" (see for instance [87, page 6]), and let χ be an observed curve. Similar nonparametric functional modeling has been proposed in [161] for the prediction of biophysical parameters from vine-leaf hyperspectral images. Such models, either for the estimation or for the classification are based on the theoretical works proposed in [87, 173].

2.2.1 Model presentation

Let C be the number of classes considered. For each class $c \in \{1, \ldots, C\}$, $p_c(\chi) := \mathbb{P}(Y = c|\mathcal{X} = \chi)$ denotes the conditional probability of belonging to the class c knowing the observed curve χ, with Y a categorical variable and $\mathcal{X}$ a random curve. Then we have $p_c(\chi) = \mathbb{E}(\mathbf{1}_{[Y=c]}|\mathcal{X} = \chi)$, with $\mathbb{E}$ the mathematical expectation and $\mathbf{1}_{[Y=c]}$ the indicatrix of $Y = c$, i.e., $\mathbf{1}_{[Y=c]} = 1$ if $Y = c$, and 0 otherwise. The functional nonparametric model is defined by a regular operator p_c, mapping the curves into the range $[0\,;\,1]$.

For the classification of individuals given χ, C kernel estimators $\hat{p}_1(\chi), \ldots, \hat{p}_C(\chi)$ are computed. The predicted class is given by the Bayes rule, affecting χ with the class $\hat{y}$ for which the estimated conditional probability membership is the highest : $\hat{y} = \underset{c\in\{1,\ldots,C\}}{\arg\max}\, \hat{p}_c(\chi)$.

$\hat{P}(\chi)$ is constructed from the pixels of the training set $\{(\chi_j, y_j)_{j=1}^n\}$, with n the training set size, following the functional Nadaraya-Watson kernel estimator formula [87, page 115] :

$$\hat{p}_c(\chi) = \frac{\sum\limits_{i \;|\; y_i=c} k(h^{-1}\ \delta(\chi, \chi_i))}{\sum\limits_{j=1}^{n} k(h^{-1}\ \delta(\chi, \chi_j))}, \tag{2.1}$$

with χ_j for $j \in \{1, \ldots, n\}$ the observed curves, k an asymmetric kernel function, h a strictly positive parameter and δ a measure of closeness between two curves.

The function k gives more weight to the observations χ_j that are close to the curve χ, according to the measure of closeness δ. It has been shown in the literature that the influence of the kernel function k on the speed of convergence of the functional Nadaraya-Watson estimator is negligible compared to the influence of the other two parameters h and δ [87] ; so the use of a quadratic asymmetric kernel k_0 such as $k_0(t) := \frac{3}{2}\ (1 - t^2)\ \mathbf{1}_{[0;1]}(t)$ was arbitrarily chosen for this estimator. h acts as a bandwidth parameter of the estimator : the k function being nonzero on $[0; 1]$ and zero elsewhere, the curve χ_i will be part of the estimator if and only if $\delta(\chi, \chi_i) < h$. Thus, a higher value of h will allow the estimator to take into account curves χ_i far from χ

according to the measure of closeness δ, giving a higher weight to closer curves, while a lower value of h will restrict the estimator to only consider curves closer from χ. As shown in section 2.3, the proximity measure δ plays a crucial role. Moreover, in order to make the model more flexible and to reduce the curse of dimensionality impact, the choice of δ is extended to some pseudometrics family. Next section presents three particular pseudometrics.

2.2.2 Pseudometrics

Several pseudometrics have been investigated in this paper. The first is the standard L^2 metric, the second uses a functional principal components analysis (FPCA) [173, 102] of the data and the third is based on the computation of partial least squares (MPLSR) [107, pages 66-68].

A pseudometric δ is nearly a metric as it satisfies three metric axioms out of four (positivity, symmetry and triangle inequality) but not necessarily the indistinguishability axiom ($\delta(\chi_i, \chi_{i'}) = 0 \nRightarrow \chi_i = \chi_{i'}$).

The first pseudometric used is the standard L^2 metric δ^{L^2} defined for all curves χ_i and $\chi_{i'}$ as :

$$\delta^{L^2}(\chi_i, \chi_{i'}) = \sqrt{\int (\chi_i(\lambda) - \chi_{i'}(\lambda))^2 \, d\lambda}.$$

This definition is the natural functional extension of the vectorial L^2 metric. In that case, δ^{L^2} is a metric because it satisfies the indistinguishability axiom ($\delta^{L^2}(\chi_i, \chi_{i'}) = 0 \Leftrightarrow \chi_i = \chi_{i'}$). δ^{L^2} uses the whole information contained in the curves and hence may suffer from the curse of dimensionality.

The second pseudometric δ^{FPCA} is based on a functional principal components analysis. This pseudometric is defined for all curves χ_i and $\chi_{i'}$ as :

$$\delta^{FPCA}_{q_1,q_2}(\chi_i, \chi_{i'}) = \sqrt{\sum_{m=q_1}^{q_2} \left(\int [\chi_i(\lambda) - \chi_{i'}(\lambda)] \nu_m(\lambda) \, d\lambda \right)^2},$$

with $\nu_1, \nu_2, \ldots$ the functions associated with the decreasing eigenvalues $\mu_1 \geq \mu_2 \geq \ldots$ of the covariance operator $\Gamma_\chi(\lambda, \lambda') = \mathbb{E}[\bar{\chi}(\lambda)\bar{\chi}(\lambda')]$, where $\bar{\chi}$ stands for the centered version of χ. q_2 is the usual FPCA parameter which controls the dimension of the decomposition ; q_1 is introduced to compensate that the first principal component represents the albedo, not discriminative in general.

The third pseudometric δ^{MPLSR} use the computation of Partial Least Squares and is defined for all curves χ_i and $\chi_{i'}$ as :

$$\delta^{MPLSR}_{q_0}(\chi_i, \chi_{i'}) = \sqrt{\sum_{m=1}^{q_0} \left(\int [\chi_i(\lambda) - \chi_{i'}(\lambda)] \nu^Y_m(\lambda) \, d\lambda \right)^2},$$

with $\nu^Y_1, \nu^Y_2, \ldots$ the eigenfunctions of the partial least squares decomposition, and q_0 the parameter which controls the dimension of this decomposition. The main difference between FPCA and MPLSR pseudometrics is due to the eigenfunctions dependance in the response variable Y for the MPLSR pseudometric construction, which is not the case for the FPCA pseudometric.

Figure 2.1 introduces first (left graphic) and second (right graphic) FPCA and MPLSR components of *MADONNA* dataset. These graphics show the difference induced by taking into

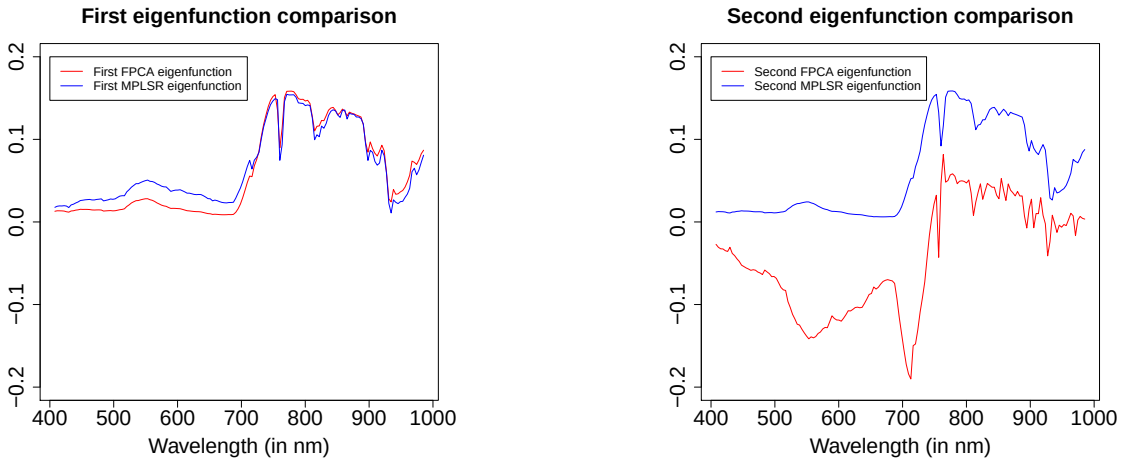


FIGURE 2.1 – Comparison of FPCA and MPLSR components.

Species	Chestnut	Walnut	Linden	Ash	Maple	Oak
Number of pixels	2855	1016	3402	4333	165	10981
Species	Fern	Hazel	Beech	Birch	Goat willow	Locust
Number of pixels	1983	4122	42	468	485	2372

TABLE 2.1 – Number of pixels associated with each studied woody species (*MADONNA* dataset).

Classes	Asphalt	Meadow	Gravel	Tree	Metal Sheet	Bare Soil	Bitumen	Brick	Shadow
Number of pixels	6631	18649	2099	3064	1345	5029	1330	3682	947

TABLE 2.2 – Number of pixels associated with each *Pavia* studied classes.

account the categorical response or not to build the functional decomposition basis.

Contrary to δ^{L^2}, $\delta^{FPCA}_{q_1,q_2}$ and $\delta^{MPLSR}_{q_0}$ aim to project the curves onto a data-driven finite-dimensional subspace. In this way, $\delta^{FPCA}_{q_1,q_2}$ and $\delta^{MPLSR}_{q_0}$ are potential candidates for reducing the curse of dimensionality impact.

2.3 Experimental results

The first studied dataset, called *MADONNA*, was collected on the site of Villelongue, France, by the HYSPEX sensors. The data consists in 32224 pixels, with 160 spectral bands (from 400 to 1000 nm), and a spatial resolution of 50 cm per pixel. 12 woody species have been identified during a field campaign. *MADONNA* classes name and available referenced pixels are given in Table 2.1.

The second one is the standard dataset from *University of Pavia*, Italy, collected by an airborne ROSIS-03 (Reflective Optics System Imaging Spectrometer) optical sensor. The data consists in 42776 pixels, with 103 spectral bands (from 430 to 860 nm), and a spatial resolution of 1.3 m per pixel, for a total of 9 classes [74]. *Pavia* classes name and available referenced pixels are given in Table 2.2.

For both datasets, 30 pixels for each class were randomly chosen to train each model and the

Model	Multinomial logistic	GMM	nonlinear SVM	Nonparametric functional		
Pseudometric				L^2	FPCA	MPLSR
Average prediction error rate	71.13%	15.75%	14.13%	53.05%	29.42%	12.96%
Standard deviation	0.0413	0.0218	0.0145	0.1158	0.0484	0.0160

TABLE 2.3 – *MADONNA* average prediction error rates and standard deviations for 50 times repeated models.

Model	Multinomial logistic	GMM	nonlinear SVM	Nonparametric functional		
Pseudometric				L^2	FPCA	MPLSR
Average prediction error rate	66.21%	16.84%	18.91%	33.64%	33.84%	21.86%
Standard deviation	0.0417	0.0216	0.023	0.0249	0.0253	0.0425

TABLE 2.4 – *Pavia* average prediction error rates and standard deviations for 50 times repeated models.

FIGURE 2.2 – Classification result for a *Pavia* image using nonparametric functional method with MPLSR pseudometric.

remaining pixels composed the test set. This process has been repeated 50 times such as a new training set has been generated for each repetition. Finally, an Analysis of Variance (ANOVA) test was applied on the 50 error rates to check if the differences observed are statistically significant. The GMM model was built with a covariance matrix regularization (the covariance matrix Σ_c was replaced by its regularized version $\Sigma_c + \mu I$), while SVM method, using a Gaussian kernel, requires the setting of the kernel parameter and the soft margin penalization. Models parameters were tuned using a 5-fold cross-validation, except h in (2.1) which was tuned using a Leave-One-Out Cross-Validation (LOOCV) [107, pages 241-245].

Average classification error rates and corresponding standard deviations for the 50 repetitions are reported in Table 2.3 for *MADONNA* and Table 2.4 for *Pavia*. A thematic map using the nonparametric functional method with MPLSR pseudometric is given in Figure 2.2.

From Table 2.3, it can be seen that the multinomial logistic model and the nonparametric functional model with the L^2 metric perform the worst in terms of error rates, while nonparametric model with FPCA pseudometric provides a middling error rate. The three best results

were obtained by GMM, SVM and nonparametric functional with MPLSR pseudometric models. The ANOVA test confirmed that the classification accuracies is significantly better for MPLSR nonparametric model than the two others. When comparing the results coming from the three pseudometrics, δ^{L^2} is the most sensitive to the curse of dimensionality, while $\delta^{FPCA}_{q_1,q_2}$ reduces this sensitivity and $\delta^{MPLSR}_{q_0}$ is the most adapted to these settings.

From Table 2.4, it can be seen that the multinomial logistic model perform once again the worst in terms of error rates, while nonparametric model with L^2 or FPCA pseudometric provides a middling error rate. The three best results were obtained by GMM, SVM and nonparametric functional with MPLSR pseudometric models, although this time, GMM and SVM seem to be slightly better than nonparametric model with MPLSR pseudometric.

For both datasets, concerning nonparametric functional model with FPCA pseudometric, majority of all repetitions selected components from the second. This ascertainment confirms the non-discriminative nature of the average albedo contained in the first FPCA component.

The R codes used to develop nonparametric functional methods are available at : http://www.math.univ-toulouse.fr/staph/npfda/.

2.4 Conclusion

In conclusion of these experiments, taking into account the functional aspect of the data and combining nonparametric modelling with pseudometrics as proximity measure is an interesting alternative for analysing hyperspectral images, especially if the considered classes lead to similar hyperspectra. Within the scope of nonparametric functional method, MPLSR pseudometric appear as a better pseudometric than FPCA and L^2 on both datasets. Aspects such as smoothing data denoising, spectral bands selection or spatial correlations inclusion will be studied later in order to expect an even better predictive model.

2.5 Acknowledgments

This research was supported in part by the French National Spacial Agency (CNES) and the Midi-Pyrénées region.

Comparison of functional and multivariate spectral-based supervised classification methods in hyperspectral image

Anthony Zullo[a,b], Mathieu Fauvel[a] and Frédéric Ferraty[b]

[a] *UMR 1201 DYNAFOR, INRA & INP Toulouse, France*;
[b] *UMR 5219 Institute of Mathematics, University of Toulouse, France*

Abstract. The aim of this article is to assess and compare several statistical methods for hyperspectral image supervised classification only using the spectral dimension. Since hyperspectral profiles may be viewed either as a random vector or a random curve, we propose to confront various multivariate discriminating procedures with functional alternatives. Six methods representing three important statistical communities (mixture models, machine learning and functional data analysis) have been applied on two hyperspectral datasets following three protocols studying the influence of size and composition of the learning sample, with or without noised labels. Besides this comparative study, this work proposes a functional extension of multinomial logit model as well as a fast computing adaptation of the nonparametric functional discrimination. As a by-product, this work provides a useful comprehensive bibliography and also supplemental material especially oriented towards practitioners.

Keywords. Functional multinomial logit model ; Hyperspectral profile ; Mixture models ; Nonparametric functional discrimination ; Random Forest ; Support Vector Machines.

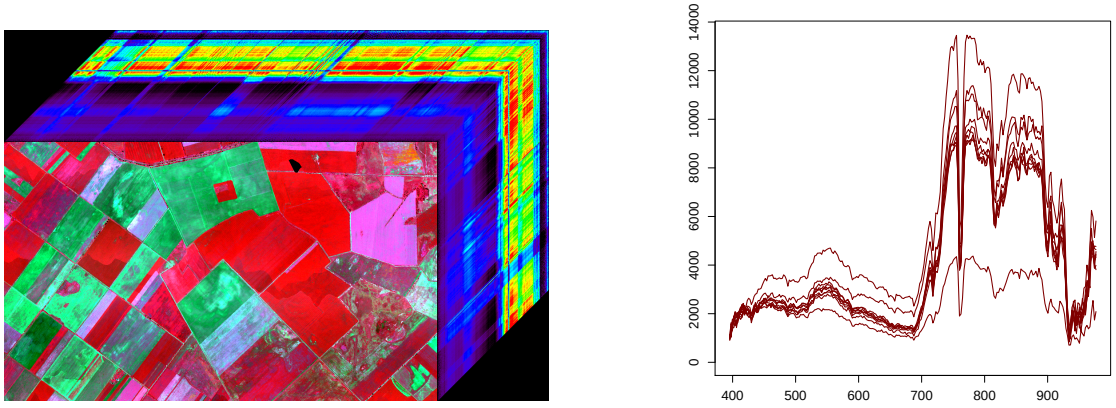
Classification codes : 62-07 ; 62H30 ; 62H35 ; 62P30.

2.6 Introduction

The use of statistical tools on remote sensing images has been recently investigated in the literature. Cardot *et al.* [33] compared two functional approaches, either an extension of the multilogit model for functional data or relying on varying-time regression models. Cardot *et al.* [35] proposed the use of varying-time random effects models for unmixing and temporal interpolation of longitudinal data. Kazianka *et al.* [125] provided a Bayesian approach for linear spectral unmixing with unknown covariance structure, applied on hyperspectral data. Majumdar *et al.* [147] analyzed the multivariate spatial distribution of plant species diversity through a non-stationary spatial generalized linear mixed model approach. Pasanen and Holmström [165] proposed an approach using Bayesian statistical modeling and simulation-based inference in order to detect land cover changes in satellite images.

In the last decade, the classification of hyperspectral images has received a lot of attention from the scientific community [77]. In a hyperspectral image, a standard observation for some pixel i leads to a d-dimensional random vector $\mathbf{X}_i = (X_i^1, \ldots, X_i^d)^T$ corresponding to a spectrum sampled at d wavelengths $\lambda^1, \ldots, \lambda^d$ (i.e., $X_i^j = X_i(\lambda^j)$ for $j \in \{1, \ldots, d\}$ and $d > 100$) and a categorical response $Y_i \in \{1, \ldots, C\}$ indicating the label among C classes of membership. Given a learning set $\{(\mathbf{X}_i, Y_i) \text{ for } i \in \{1, \ldots, n\}\}$, the classification problem consists in assigning each pixel of the image to a class. Hyperspectral images are standardly displayed as 3-D objects (see Figure 2.3 (a)) : two dimensions for the spatial coordinates plus a third dimension along the available wavelengths.

Two hyperspectral datasets will be studied in this article. The first is the *University of Pavia* (Italy) dataset, with spectra sampled at $d = 103$ wavelengths, while the second, called *AISA*, involves $d = 252$ wavelengths. These examples are complementary because they present significant differences in terms of spectral and spatial dimension as well as the nature of their classes.



(a) (b)

FIGURE 2.3 – A hyperspectral cube from the *AISA* dataset (a) with two spatial and one spectral dimensions, and some extracted hyperspectra (b) from the class 'Broadleaved Forest'.

The *University of Pavia* dataset contains urban (road, metal sheet, ...) and vegetation classes while the *AISA* dataset contains vegetation features only. Hence, the supervised classification task is more difficult for the *AISA* dataset than for the *Pavia* one.

Used in various application fields (food safety, pharmaceutical process monitoring and quality control, biomedical, industrial, biometric, forensic, ...), hyperspectral images have been analyzed following different methology (data fusion, dimensionality reduction, feature mining, ...), in order to achieve distinct objectives (restoration, unmixing, classification, segmentation, target detection, physical parameter retrieval, ...) [11]. Several statistical methods have already been applied on hyperspectral images. For instance, Principal Component Analysis (PCA) projection has been applied on hyperspectral data for feature extraction in the spectral domain [46], dimension reduction [72], compression [63] as well as for classification [169].

The supervised classification of hyperspectral images is a nontrivial task ; most of conventional discriminating methods are not appropriate to such datasets [121]. The failure of these methods is caused by the combination of two main features. The first is related to the large number d of spectral variables of sampled spectra when only small learning sample size is available, which is a common situation in practice. This unconfortable context, known under the terminology *curse of dimensionality* [62], leads to a drop of most standard classifiers predictive performance. Several works have been done to circumvent this issue, using Bayesian models [133], feature extraction and feature reduction techniques [133, 26], Random Forest [104], neural networks [175] or kernel methods [30]. Among these methods, Support Vectors Machines (SVM) have shown very good performances in terms of classification accuracy. A second particular feature of hyperspectral data is the existence of high correlations between consecutive variables (i.e., X_i^j and X_i^{j+1}), inducing for instance ill-conditioned covariance matrix. One way to work around such statistical problem consists in considering the i^{th} sampled hyperspectrum $\mathbf{X}_i$ as a discretized version of the curve $\chi_i := \{\chi_i(\lambda);\ \lambda \in [\lambda_{\min}, \lambda_{\max}]\}$, where the word *curve* stands for a real quantity $\chi_i(\lambda)$ varying continuously with the wavelength λ. Figure 2.3 (b) illustrates the functional (continuous) nature of hyperspectra. Formalizing the hyperspectra as curves enjoys the advantage to consider intrinsic feature like the order of the wavelengths or the shape of the hyperspectra profiles. This framework leads to the use of so-called functional data analysis methodology [172, 173], taking into account this functional nature of hyperspectra.

According to their functional aspect, hyperspectra are quite rich objects and a natural question arises : how to extract information of hyperspectral profiles for discriminating pixels in

some optimal way ? In order to answer this question, we firstly drop the spatial information (i.e., location of pixels) and focus on the spectral domain only. In other words, classifying pixels amounts to discriminate spectral profiles. Secondly, we propose to assess and compare a wide scope of multivariate or functional spectral-based supervised classification methods, which is the main contribution of the paper. An additional main interest of this work is then to identify the more relevant methods in various experimental conditions, especially for a small sampled noised learning set framework. A representative panel of six supervised classification methods is considered, benchmark procedures as well as more recent ones : Mixture models (Ridge regularized Gaussian Mixture Models and High-Dimensional Discriminant Analysis), machine learning methods (Support Vector Machines and Random Forest) and functional data analysis methods (Functional Multinomial Logit Model and NonParametric Functional Discrimination). All methods have been implemented and compared on both datasets following different protocols in order to assess their behaviour in various situations combining learning set size, composition of in-sample and way of noising.

In addition of this comparative study, this work also proposes a functional extension of the Multinomial Logit Model as well as a fast computing adaptation of the NonParametric Functional Discrimination method. Another interesting by-product of this work is the providing of a useful wide bibliography on the selected sample of classification methods as well as supplemental material allowing to implement in an easy way this comparative study.

In the following, classification methods are described in Section 2.7 systematically accompanied by a comprehensive bibliography. In Section 2.8, a comparison of these methods on two real datasets is provided. Experimental protocols as well as implementation and results are also presented. Section 2.9 is devoted to discussion and concluding remarks.

2.7 Representative classification methods

This section describes the selected panel of supervised classification methods that are discussed in the paper. The following notations are defined for the remaining of this article : when considering multivariate discriminating methods, $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are samples of d-dimensional random vectors containing the discretized hyperspectra ; in functional context (i.e., hyperspectra profiles are considered as a collection of curves), $\mathcal{X}_1, \ldots, \mathcal{X}_n$ stand for samples of the random curve $\mathcal{X} := \{\mathcal{X}(\lambda);\, \lambda \in [\lambda_{\min}, \lambda_{\max}]\}$, a particular case of *functional variable* in the functional data analysis community (see for instance [87]). At last, $\langle \cdot, \cdot \rangle$ (resp. $\| \cdot \|$) stands for any inner product (resp. norm).

2.7.1 Mixture Models

Mixture models are probabilistic methods used to cluster data into several groups [154]. This family of models assumes all vectors $\mathbf{X}_i$ to be distributed following a probabilistic density function g, a mixture of several probability functions $f_1, \ldots, f_C$ conditionally to the class c in such a way that $g(\mathbf{X}_i) = \sum_{c=1}^{C} \pi_c \; f_c(\mathbf{X}_i)$, where π_c are unknown weights. These functions $f_1, \ldots, f_C$ are usually assumed to be the same probability law which parameters are unknown and are to be estimated. In a supervised classification context, using the maximum a posteriori and the Bayes rule, the response variable y_i is then estimated as $y_i = \underset{c \in \{1,\ldots,C\}}{\arg\max} \{\pi_c \; f_c(\mathbf{X}_i)\}$.

2.7.1.1 Gaussian Mixture Models (GMM)

As a simple mixture model, the Gaussian Mixture Model [106], also called Quadratic Discriminant Analysis, is widely used in many application fields. It is assumed that each vector, conditionally to class $c \in \{1, \ldots, C\}$, follows a multidimensional Gaussian probability law, i.e.,

$f_c \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$, where $\boldsymbol{\mu}_c$ stands for the mean vector and $\boldsymbol{\Sigma}_c$ is the covariance matrix of the Gaussian law. The Gaussian mixture decision rule is :

$$y_i = \underset{c\in\{1,\dots,C\}}{\arg\max} \left\{ -\left(\mathbf{X}_i - \boldsymbol{\mu}_c\right) \boldsymbol{\Sigma}_c^{-1} \left(\mathbf{X}_i - \boldsymbol{\mu}_c\right)^T - \ln\left(|\boldsymbol{\Sigma}_c|\right) + 2\ln(\pi_c) \right\}. \tag{2.2}$$

$\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ are usually estimated using empirical estimators.

In a high-dimensional context, two potential issues of GMM are known : the estimation of the determinant and of the inverse of covariance matrices $\boldsymbol{\Sigma}_c$. Indeed, GMM suffers from the increasing size of the covariance matrix, causing in practice a bad conditioning for inverse and determinant computations that lead to numerically unstable results [166]. Sub-models derived from the original GMM act on the definition and the estimation of covariance matrices $\boldsymbol{\Sigma}_c$. In order to keep a general and flexible framework, Celeux and Govaert [38] provide several parsimonious cluster-based GMM forcing equality constraints between coefficients of covariance matrices decompositions. A simple way to avoid models unstability is to consider the Ridge regularization of the covariance matrix $\boldsymbol{\Sigma}_c$. Thus, a ridge regularized version of the Gaussian mixture (RGMM) can be derived by replacing the covariance matrix $\boldsymbol{\Sigma}_c$ with $\boldsymbol{\Sigma}_c + \nu\mathbf{I}$ in the equation (2.2), where ν is a positive parameter to be tuned and $\mathbf{I}$ stands for the identity matrix. Several other methods have been investigated so far to overcome these computational issues. An overview of mixture models for high-dimensional data is provided in Bouveyron and Brunet-Saumard [17], where dimension reduction, regularization, constraints and parsimony, as well as variable selection and subspace clustering methods are discussed.

Some high-dimensional GMM have been developed and applied to hyperspectral imaging. In Landgrebe [133, chap. 4] is provided a hybrid covariance estimation method, especially well adapted when the size of the learning set is small compared to the number of spectral variables. Based on a mixture of various types of covariance matrices, this approach provided slightly better results than estimators using a Leave-One-Out Cross-Validation (LOOCV) on both simulated and real hyperspectral datasets. Using Cholesky decomposition, Berge *et al.* [10] and Jensen *et al.* [119] estimated the inverse covariance matrix using a sparse approximation.

2.7.1.2 High-Dimensional Discriminant Analysis (HDDA)

Providing better results than the standard GMM approach, Bouveyron *et al.* [19] presented a method called High-Dimensional Discriminant Analysis (HDDA), based on the assumption that high-dimensional data live in different subspaces with low dimensionality. Combining dimension reduction and constraints on the model, this approach is a new parametrization of GMM based on a covariance assumption such as eigenvalues can be separated in two groups in order to achieve a signal/noise covariance matrix decomposition and reduce the number of parameters to estimate.

2.7.2 Machine learning methods

We propose in this section to focus on two popular successfull methods coming from the machine learning community : Support Vector Machines and Random Forest.

2.7.2.1 Support Vector Machines (SVM)

Introduced in Vapnik [209], Support Vector Machines (SVM) are nonparametric binary classifiers, trying to find the best hyperplane separating observations into two parts while being as far as possible from them. Without loss of generality, both classes can be arbitrarily relabeled as -1 and 1 (then $y \in \{-1; 1\}$). To be able to perform nonlinear classification, SVM use the

so-called 'Kernel trick', defining a reproducing kernel function K [110]. Then, for any given kernel K, the SVM decision rule can be written as $y_i = \text{sign}\left\{\sum_{i'=1}^{n} \alpha_{i'}\, y_{i'}\, K(\mathbf{X}_i, \mathbf{X}_{i'}) + \beta\right\}$, where $\alpha = (\alpha_1, \ldots, \alpha_n)$ and β are unknown real parameters, usually estimated through constrained quadratic optimization [71]. In practice, common kernels are $K(\mathbf{X}_i, \mathbf{X}_{i'}) = (\langle \mathbf{X}_i, \mathbf{X}_{i'} \rangle + 1)^p$, $p \in \mathbb{N}^*$ (polynomial kernel) or $K(\mathbf{X}_i, \mathbf{X}_{i'}) = \exp(-\gamma ||\mathbf{X}_i - \mathbf{X}_{i'}||^2)$, $\gamma \in \mathbb{R}_+^*$ (Gaussian kernel).

SVM binary classifiers can be extended for multiclass purpose, by breaking multiclass problems into several binary ones. Also called *pairwise classification methods*, two opposite approaches called *One-Versus-All* and *One-Versus-One* are mainly used [128, 42].

There exists a wide literature on SVM and kernel methods applied on various fields [110]. Less affected by the high dimensionality, SVM is widely used in hyperspectral imaging and other scientific applications using high-dimensional data. Mountrakis *et al.* [157] expose a review of SVM and many variations and extensions in the remote sensing field. SVM is presented as a method more relevant than other standard classifiers in a small training set context. This is especially the case in Furey *et al.* [90], where SVM classification and feature selection (among thousands of variables) applied on microarray expression data are presented. Several SVM pool-based active learning algorithms using high-dimensional datasets are proposed in Tong and Koller [205] for text classification. This approach consists in iteratively labeling learning samples located in the SVM margin and reconstructing the hyperplane on both labeled and unlabeled learning samples. These algorithms substantially outperform the standard passive learning but are much more computationally intensive.

Conventional SVM has been widely applied in the remote sensing field for the classification of hyperspectral datasets [31, 155, 163]. Several specific adaptations for hyperspectral classification have been investigated, as in Mercier and Lennon [156] where kernels are built in replacing the Euclidean distance by distances based on spectral information.

SVM can also be extended to functional classification, using kernels taking into account the functional nature of the data. Rossi and Villa [179] presented an approach based on functional data projections on finite dimensional spaces, then using the standard multivariate SVM on projected coordinates. The choice of the projection subspace can be directed by expert knowledge on functions (e.g., Fourier, wavelet or B-spline basis). Park *et al.* [164] aim to detect batteries faults thanks to functional SVM. This adaptation uses first and second derivatives of degradation profiles, based on a B-spline approximation to avoid numerical unstability problem of direct computation. A feature selection is then applied on these sampled derivatives, leading to the use of the standard multivariate SVM.

2.7.2.2 Random Forest

Random forests are nonparametric classification methods based on the construction of a large collection of decision trees. Decision trees are nonlinear predictive models which, in a supervised classification context, are built on the learning set using recursive partitioning. The Classification And Regression Trees (CART) algorithm [22] builds each tree node in minimizing a splitting criterion called Gini impurity. However, CART algorithm can suffer from some drawbacks. Based on heuristics, CART is a greedy algorithm, then there is no guarantee of finding the globally optimal decision tree from locally optimal splits. Some methods are able to overcome this issue (e.g., [9]). CART algorithm is also subject to overfitting. A pruning mechanism can be necessary to stop splitting if the provided improvement is not considered as significant, reducing the final decision tree complexity and achieving a bias-variance trade-off.

To stabilize the decision tree method, the idea of building a lot of trees leads to the so-called *Forest* notion [21]. Random Forest tree collection is usually built using Bootstrap aggregating (also called Bagging) [20] and random feature selection, reducing individual trees correlations and providing robustness to overfitting. The pruning mechanism is then no more needed as

the high variance provided by each fully developed tree is reduced by combining them as a Forest. Random Forest classifier predicts the categorical response variable y as the majority vote provided by individual trees. This method have been successfully applied in various domains including remote sensing, such as in Pal [162] for multispectral land cover classification and in Gislason *et al.* [96] for the classification of multisource remote sensing and geographic data.

Unlike other standard multivariate methods, Random Forest can take advantage from high-dimensional data as it does not suffer from the curse of dimensionality, as explained in Schwarz *et al.* [183] where a fast implementation of Random Forest method for high-dimensional data is exposed. An application of Random Forest for microarray-based cancer classification is provided in Statnikov *et al.* [190]. However, when variables are strongly correlated, Random Forest (among other methods) suffers from a bias. This issue can be solved using group selection methods based on feature clustering [204].

Hyperspectral data have been studied using Random Forest, for instance in Lawrence *et al.* [135] for invasive plants mapping. Chan and Paelinckx [39] compared Random Forest with an alternative tree-based ensemble method for classification and spectral band selection of hyperspectral images. Applied on an airborne hyperspectral dataset for ecotope mapping, both methods attain almost the same overall accuracy, however Random Forest is faster in terms of computation time with less accuracy variability. In Ham *et al.* [104] is provided an approach based on Random Forest of complex binary trees for hyperspectral data classification. As an alternative approach to the CART-based framework, these complex binary trees are decision trees based on an iterative decomposition of the multiclass problem into nested binary classifications of classes groups. Compared on three hyperspectral datasets, Random Forest of complex binary trees gave consistently better results than related methods, especially for smaller training set sizes, however with a much higher computational cost.

Random Forest has also been applied on functional data thanks to reproducing kernels. As an example, Kernel-Induced Random Forest (KIRF) method has been extended in Fan *et al.* [68]. Kernel-induced classification trees are built using a kernel applied on each training sample pairs as candidate splitting rules. The key point of the KIRF functional extension is the definition of kernel functions applied to two functions, estimated from noisy data using nonparametric smoothing methods, such as functional PCA [173, chap. 8] or penalized spline smoothing. Functional KIRF method provided better results in terms of classification error rate compared to other functional methods on a temporal gene expression data.

2.7.3 Functional methods

Functional methods are a family of statistical methods especially built to handle functional data. The use of a functional representation offers the benefit of taking into account features not supported by multivariate methods, like the continuous nature of hyperspectra as well as the order and correlations of spectral bands. Two ways are mostly used to handle functional data. One way consists in operating a basis expansion of the functional data, then applying standard multivariate methods on the coefficients of this decomposition. The other way keeps the functional framework of the data, applying on it approximates of functional operators.

2.7.3.1 Generalized Linear Models and functional extension

Generalized Linear Models (GLM) are extensions of the ordinary linear regression using an invertible known link function g [152]. The most used GLM for classification is called Logistic Model (or Logit Model), which link g is the logit function defined on $]0,1[$ as $g(p) = \log\{p/(1-p)\}$. In a classification context, any GLM can be written for all $c \in \{1,\dots,C\}$ and for all $i \in \{1,\dots,n\}$ as $\mathbb{P}(y = c|\mathbf{X} = \mathbf{X}_i) = g^{-1}(\langle \mathbf{X}_i^*, \beta_c \rangle)$, where $\beta_1,\dots,\beta_C$ are vectors of size $d+1$ contai-

ning the unknown parameters and $\mathbf{X}_i^* := (1, X_i^1, \ldots, X_i^d)^T$. Originally created to discriminate two classes, the Logit Model has been naturally extended for multiclass purpose as the so-called Multinomial Logit (or Logistic) Model, which can be written as $\mathbb{P}(y = c|\mathbf{X} = \mathbf{X}_i) = \exp\langle \mathbf{X}_i^*, \beta_c \rangle / \sum_{c_0=1}^{C} \exp\langle \mathbf{X}_i^*, \beta_{c_0} \rangle$, setting $\langle \mathbf{X}_i^*, \beta_c \rangle = 0$ for one $c \in \{1, \ldots, C\}$. The class y_i is then predicted following the Bayes rule, affecting each individual vector $\mathbf{X}_i$ the class which estimated membership probability is the highest : $y_i = \underset{c \in \{1,\ldots,C\}}{\arg\max}\, \mathbb{P}(y = c|\mathbf{X} = \mathbf{X}_i)$.

There is an abundant literature on Multinomial Logit Model applications such as in van Rees *et al.* [176] on people reading behavior, in Choo and Mokhtarian [50] on consumers' car buying behaviors, in Stratton *et al.* [192] on college stopout and dropout behavior or in Chiswick and Miller [49] on people educational level and occupational attainment.

In a high-dimensional context, neither the least squares estimator nor the maximum likelihood can be standardly used because the covariance matrix becomes ill-conditioned. Several approaches were proposed to overcome this issue, using a Ridge regularization or a Lasso penalisation [57]. Krishnapuram *et al.* [129] developed fast algorithms to implement a sparse version of the Multinomial Logistic Regression based on a maximum a posteriori approach with a Gaussian or a Laplacian prior, equivalent to a Ridge or a Lasso penalisation respectively. Based on this approach, the Multinomial Logit Model has also been applied on hyperspectral images like in Li *et al.* [138] where a semi-supervised active learning segmentation algorithm is exposed. Based on a Multinomial Logit Model built from both labeled and unlabeled training samples, this algorithm uses both spectral and spatial information in a Bayesian approach. Li *et al.* [141] presented a new framework for hyperspectral image classification by Multinomial Logistic Regression. Generalizing composite kernels to combine spectral and spatial information, this approach uses a Laplacian prior to build a sparse Multinomial Logistic model. A subspace-based Multinomial Logistic Regression method is proposed in Khodadadzadeh *et al.* [126] for pixelwise hyperspectral image classification. This model includes class prior probabilities in linear combinations of feature vectors obtained by subspace projections.

An adaptation appears necessary to apply GLM on usually strongly correlated functional predictors [117, 159]. The Logit Model has been adapted to solve functional problems, as in Escabias *et al.* [66] where a Functional Logit Model based on a Functional PCA (FPCA) is applied on a climatological data. Aiming to address multicollinearity and high dimensionality problems, this approach uses a cubic spline interpolation. Leng and Müller [136] used the functional data analysis for the classification of temporal gene expression data. The Functional Logit Model based on FPCA provided slightly lower classification error rates while employing fewer components than a B-spline implementation of functional discriminant analysis on both real and simulated datasets. Müller [158] proposed an extension of the Generalized Functional Linear Model to the case of noisy, sparse and irregular data, applied on longitudinal predictors.

According to the current literature, functional extension are available only for two-classes purpose. That is why we propose our own functional extension of the Multinomial Logit Model (FMLM), expanding the functional data by means of a spline basis in order to apply the Multinomial Logit Model on the coefficients of this decomposition.

2.7.3.2 NonParametric Functional Discrimination (NPFD)

Nonparametric statistics refers to a set of distribution free data-driven methods, in the sense of methods which do not rely on any parametric assumption nor any given probability distribution. The use of nonparametric models leads to more general, robust, flexible and adaptable models, while allowing the exploration of nonlinear relationships. In this high-dimensional context, nonparametric classification methods can be built in several ways, as in Greenshtein and Park [99] where a Nonparametric Empirical Bayes Estimation is proposed. This approach combines

many *weakly informative* variables using naive Bayes classifiers. In a hyperspectral classification context, several recent articles report the use of nonparametric methods, as in Delalieux *et al.* [59] for the detection of biotic stress in apple trees. The use of nonparametric functional methods on hyperspectral datasets have only been recently investigated, as in Ordóñez *et al.* [161] for vine-leaf composition characterization.

Nonparametric functional discrimination belongs to the general nonparametric functional data analysis framework proposed in Ferraty and Vieu [87]. In our hyperspectral classification context, the nonparametric functional model can be defined for each class $c \in \{1, \ldots, C\}$ by a regular operator $r_c(\chi_i) := \mathbb{P}(y = c|\mathcal{X} = \chi_i)$, mapping an observed hyperspectrum χ_i at pixel i into the range [0 ;1]. Only smoothness hypothesis about these C operators $r_1, \ldots, r_C$ are required by this approach. Remark that $\mathbb{P}(y = c|\mathcal{X} = \chi_i) = \mathbb{E}\Big(\mathbf{1}_{[y=c]}|\mathcal{X} = \chi_i\Big)$. Therefore, for $c = 1, \ldots, C$, the conditional probability $\mathbb{P}(y = c|\mathcal{X} = \chi_i)$, or equivalently the operator r_c, can be estimated using a functional adaptation of the Nadaraya-Watson kernel estimator [160, 213] : $\sum_{i'=1}^{n} \mathbf{1}_{[y_{i'}=c]}\ K\Big(\frac{\delta(\chi_i,\chi_{i'})}{h}\Big) \Big/ \sum_{i'=1}^{n} K\Big(\frac{\delta(\chi_i,\chi_{i'})}{h}\Big)$, where h is a smoothing parameter, K is an asymmetric kernel function defined on $[0; 1]$ and δ is a proximity measure between two functions, which choice has been extended to some pseudometrics family. By this way, this classification estimation problem can be written as a regression estimation problem. Once each conditional probability $\mathbb{P}(y = c|\mathcal{X} = \chi_i)$ is estimated for $c = 1, \ldots, C$, the class y_i is predicted following the Bayes rule, assigning χ_i to the class which estimated membership probability is the highest : $y_i = \underset{c\in\{1,\ldots,C\}}{\arg\max}\, \mathbb{P}(y = c|\mathcal{X} = \chi_i)$. In practice, the choice of the scale parameter h is replaced by an equivalent choice of a k-nearest neighbours parameter k, changing a continuous parameter by a discrete one, in order to facilitate the implementation of this method.

This nonparametric functional approach has been applied in several studies, as in Ferraty and Vieu [85] on food industry and speech recognition benchmark datasets, or in Tarrío-Saavedra *et al.* [198] to classify wood species from thermal data. A comparison of nonparametric functional discriminant analysis with Fisher's linear discriminant analysis and several neural networks is provided in López-Granados *et al.* [144] for crops and weeds hyperspectral classification. The original nonparametric functional approach has been extended in Timmermans *et al.* [203], where the wavelet-based *BAGIDIS* (*BAsis GIving DIStances*) pseudometric allows to address the problem of locally sharp features.

2.8 Comparison on hyperspectral datasets

2.8.1 Datasets and experimental protocols

In this article, two hyperspectral datasets are investigated. The first is the *University of Pavia* (Italy) dataset, collected by an airborne ROSIS-03 optical sensor with 103 spectral bands ranging from 430 to 860 nm, and a spatial resolution of 1.3 m per pixel. The reference dataset consists in 42,776 pixels for a total of 9 classes including urban, soil and vegetation features [74]. The second, called *AISA*, was collected over an area containing arable lands near the city of Heves in Hungary. It has been formed using an AISA Eagle instrument, with 252 spectral bands ranging from 395 to 975 nm, and a modified spatial resolution of 6 m per pixel (originally 2 m per pixel). The reference dataset contains 94,232 pixels for a total of 15 classes [210]. Ground truth for both datasets are provided in Figure 2.4. *Pavia* and *AISA* classes names and available referenced pixels are given in Tables 2.5 and 2.6 respectively. Averaged hyperspectra over each class are represented for both datasets in Figure 2.5. According to the averaged hyperspectra shapes, both datasets reveal classification problems of different nature. The *University of Pavia* dataset is actually composed of rather heterogeneous classes (urban, soil, vegetation) while the

(a) (b)

FIGURE 2.4 – Ground truth of *Pavia* (a) and *AISA* (b) datasets. Each color corresponds to a class, while white means an absence of data.

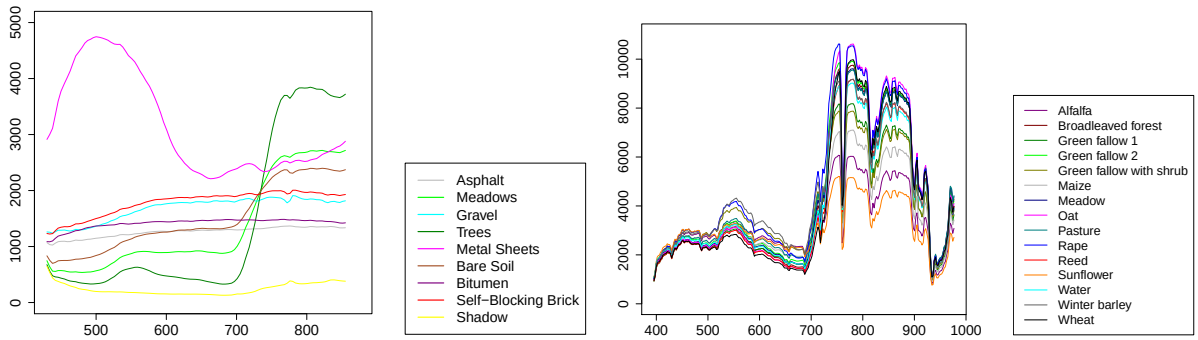
TABLE 2.5 – Number of pixels associated with each *Pavia* studied classes.

Classes	Asphalt	Meadows	Gravel	Trees	Metal Sheets	Bare Soil	Bitumen	Self-Blocking Bricks	Shadow
# pixels	6631	18649	2099	3064	1345	5029	1330	3682	947

TABLE 2.6 – Number of pixels associated with each *AISA* studied classes.

Classes	Alfalfa	Broadleaved forest	Green fallow 1	Green fallow 2	Green fallow with shrub	Maize
# pixels	5885	9308	10110	3442	3581	12602

Classes	Meadow	Oat	Pasture	Rape	Reed	Sunflower	Water	Winter barley	Wheat
# pixels	3754	3804	2094	8846	4772	6151	3417	2801	13665



(a) (b)

FIGURE 2.5 – (a) *Pavia* and (b) *AISA* averaged hyperspectra per class. The horizontal axis represents the wavelength in nanometer and the vertical axis the numerical count.

AISA dataset only contains vegetation-based features.

Three experimental protocols have been applied on each dataset. The first protocol studies the influence of a balanced learning sample size on the classification performance. n_l=30, 120, 480 pixels have been randomly chosen to train each method whereas the remaining pixels compose the test set. It corresponds to the three situations $n_l << d$, $n_l \approx d$ and $n_l > d$, d being the number of spectral bands. The second protocol provides the case of unbalanced learning set. Two approaches are compared ; the number of learning pixels per class is either proportional to the corresponding total number of pixels available in the class, or proportional to its corresponding average standard deviation over all spectral bands. Both cases have been calibrated to contain the same number of learning pixels (either $30 \times C$ or $480 \times C$). The third protocol compares the robustness of the methods to noise in learning labels. We chose the framework of Bouveyron and Girard [18] with 30 and 480 pixels per class, noising learning labels uniformly over the remaining $C-1$ classes with probability η, with η=0.15 or 0.30. Each learning label can stay unchanged with probability $1-\eta$, or be changed as one of the $C-1$ other classes with probability $\eta/(C-1)$ for each. The last case of the first protocol (480 learning pixels for each class) was chosen as the reference to compare with all others. These three protocols have been repeated 50 times such as a new training set has been generated for each repetition.

2.8.2 Implementation

Six classification methods have been compared on both datasets following these three different experimental protocols : Ridge regularized Gaussian Mixture Models (RGMM), High-Dimensional Discriminant Analysis (HDDA), Support Vector Machines (SVM), Random Forest, Functional Multinomial Logistic Model (FMLM) and NonParametric Functional Discrimination (NPFD). To apply these six statistical methods on both hyperspectral datasets, the R software [170] has been chosen. More details about computations are available in supplemental material **R_computation_details**.

2.8.3 Classification results

Average classification error rates and standard deviations for the 50 repetitions are reported in Table 2.7 for both *Pavia* and *AISA* datasets. For each configuration, results significantly better than the others (by means of analysis of variance) are highlighted in boldface.

The results of the first protocol show that the misclassification rates decrease while the learning set size increases, whatever the considered method. The second protocol results suggest that regardless of the method employed, a learning set following the total available proportional rule provides a significantly lower misclassification rate than an average standard deviation proportionality or a balanced scheme. The learning set composition also shows its low impact on methods performances hierarchy. The third protocol illustrates the predictive power deterioration of statistical models when the labels noising ratio increases, degrading the misclassification rate of all methods. According to these results, some methods seem to outperform the others in reducing the impact of noised learning labels. For instance, the misclassification rates computed from RGMM, HDDA and FMLM degrade significantly with η whereas predictive performances of SVM, Random Forest and NPFD remain very stable up to $\eta = 0.30$.

In this study, SVM is the method providing overall best results on the *Pavia* dataset. However, in a small training set size context, RGMM seems to be more relevant than SVM, this latter still providing acceptable results. NPFD also provides better results in a small sized and noised learning set framework. Several methods have already been successfully applied on this dataset, mostly in combining both spectral and spatial information. Some of these methods are SVM-based [194, 196, 197, 44]. However, most studies compare methods on the same learning-test sets without any randomization nor any repetition. The framework proposed in this article

TABLE 2.7 – Study of the learning set size, composition and noising influences on the performances of six supervised classification methods applied on both *Pavia* and *AISA* datasets ; 'RF' stands for Random Forest. This table reports average classification error rates and between brackets standard deviations expressed in percentages for 50 independent repetitions of each protocol. 'n_l' refers to the average number of learning pixels taken from each class. 'Balanced' means the same number of pixels is taken from each class. 'S-prop.' and 'D-prop.' refer to total Size (constant ratio number of learning pixels on number of test pixels among classes) or standard Deviation (constant ratio number of learning pixels on spectral average bandwise standard deviation among classes) propotionality respectively. 'Weak' and 'Strong' noises refer to the noising level η being 0.15 and 0.30 respectively. 'Noiseless' means $\eta = 0$.

Data set	n_l	Proportion	Noise	RGMM	HDDA	SVM	RF	FMLM	NPFD
Pavia	30	Balanced	Noiseless	**17.2 (2.2)**	19.7 (3.1)	18.9 (2.2)	30.8 (2.8)	27.5 (2.7)	22.0 (3.4)
	120	Balanced	Noiseless	**11.0 (1.3)**	17.8 (1.8)	**11.0 (1.0)**	21.7 (1.3)	20.2 (1.2)	13.1 (1.1)
	480	Balanced	Noiseless	8.2 (0.4)	16.6 (1.3)	**7.3 (0.4)**	14.9 (0.9)	15.4 (0.6)	9.4 (0.4)
AISA	30	Balanced	Noiseless	35.3 (1.4)	36.2 (2.3)	37.2 (1.3)	40.0 (1.2)	38.6 (1.6)	**31.1 (1.1)**
	120	Balanced	Noiseless	33.1 (0.8)	34.6 (1.4)	28.9 (0.6)	33.0 (0.5)	31.4 (1.0)	**23.8 (0.7)**
	480	Balanced	Noiseless	33.1 (0.8)	35.4 (2.2)	20.8 (0.4)	27.6 (0.4)	27.4 (0.3)	**18.8 (0.3)**
Pavia	30	S-prop.	Noiseless	14.4 (2.7)	23.1 (2.3)	**13.5 (1.3)**	20.5 (0.8)	21.4 (1.7)	16.6 (1.4)
	480	S-prop.	Noiseless	6.9 (0.2)	14.2 (1.0)	**5.6 (0.2)**	9.9 (0.3)	11.2 (0.3)	8.8 (0.3)
	30	D-prop.	Noiseless	**18.8 (3.2)**	39.0 (4.8)	**18.9 (2.3)**	33.1 (2.1)	28.9 (2.1)	23.0 (3.2)
	480	D-prop.	Noiseless	8.2 (0.4)	16.9 (1.1)	**7.5 (0.4)**	16.4 (0.8)	16.5 (0.9)	10.9 (0.6)
AISA	30	S-prop.	Noiseless	34.3 (1.0)	38.9 (5.9)	35.4 (0.9)	38.9 (0.9)	35.6 (1.5)	**28.8 (1.0)**
	480	S-prop.	Noiseless	29.8 (0.5)	32.5 (1.3)	19.0 (0.3)	24.7 (0.3)	25.0 (0.2)	**16.6 (0.2)**
	30	D-prop.	Noiseless	35.2 (0.9)	39.6 (4.9)	36.8 (1.2)	40.1 (1.1)	37.7 (1.5)	**30.0 (1.2)**
	480	D-prop.	Noiseless	30.4 (0.9)	33.6 (1.3)	19.8 (0.3)	26.0 (0.3)	26.3 (0.3)	**17.8 (0.2)**
Pavia	30	Balanced	Weak Noise	32.4 (6.1)	37.9 (6.3)	25.5 (4.7)	32.7 (3.3)	34.7 (3.6)	**23.0 (3.8)**
	480	Balanced	Weak Noise	20.3 (1.9)	37.6 (4.4)	**8.4 (0.4)**	15.3 (1.0)	22.7 (0.7)	10.7 (0.5)
	30	Balanced	Strong Noise	41.0 (7.3)	48.7 (6.4)	31.2 (6.1)	36.9 (3.5)	40.7 (5.8)	**28.0 (6.5)**
	480	Balanced	Strong Noise	25.4 (2.6)	46.8 (4.2)	**9.7 (0.7)**	16.4 (1.1)	25.7 (0.9)	11.4 (0.7)
AISA	30	Balanced	Weak Noise	39.4 (2.4)	39.4 (2.0)	39.8 (1.4)	41.4 (1.4)	41.9 (1.8)	**31.2 (1.1)**
	480	Balanced	Weak Noise	35.9 (0.9)	41.9 (2.6)	23.2 (0.4)	27.9 (0.3)	30.8 (0.6)	**19.2 (0.8)**
	30	Balanced	Strong Noise	43.8 (1.7)	44.0 (2.3)	43.4 (1.9)	44.5 (1.8)	46.0 (2.0)	**31.5 (1.1)**
	480	Balanced	Strong Noise	37.2 (1.3)	44.4 (2.2)	26.4 (0.5)	28.6 (0.4)	33.6 (0.8)	**20.3 (0.3)**

thus avoid the bias generated by such a protocol.

About the *AISA* dataset, best results are provided using NPFD with significative improvements from 2% to 12% compared to all other studied methods, depending on the protocol used.

Three methods out of the six proposed (NPFD, RGMM and SVM) seem to be adapted for hyperspectral images classification, depending on the learning framework considered. When classes are of different natures, SVM seems to be more relevant, except in a small learning set framework. RGMM provides better results in this latter case, except when labels are noised, for which NPFD seems to be more relevant. Compared to all other methods, NPFD also seems particularly efficient when classes are less heterogeneous, whatever the framework considered.

2.9 Discussion and conclusion

This comparative study results in some interesting conclusions. Firstly, multivariate and functional methods provide useful complementary approaches. Secondly, the predictive performance depends on the nature of the classification problem. In the more favorable situation (i.e.,

heterogeneous classes with no noised learning labels), multivariate supervised classification procedures SVM and RGMM provide best misclassification rates whereas in the worst setting (i.e., less heterogeneous classes with noised labels and small learning sample size) the nonparametric functional method NPFD outperforms its competitors. So, there is some advantage to consider a functional approach, especially when hyperspectral profiles outlines small differences from one class to another one or when the presence of noised labels is suspected. In this case, considering hyperspectra as curves allows to fully exploit their functional nature.

According to the obtained results, it is more challenging to discriminate the *AISA* dataset than the *Pavia* one. However, besides the apparent homogeneity of categories, the *AISA* dataset also contains mixed pixels due to its modified spatial resolution [210]. Thus, this could partly explain why the *AISA* dataset is overall less well classified by all studied methods than the *University of Pavia* dataset. Nevertheless, a natural way to reduce the misclassification rate, and this for both datasets, could consist in combining spectral and spatial information. However, as explained in the introduction, taking into account the spatial component is out of the scope of this work.

At last, as pointed out in the introduction, considering a large number of spectral variables with a small training sample size leads to the so-called *curse of dimensionality*. Studying the influence of the learning set size corroborates this mechanism, as all methods give better results with an increasing learning set size.

2.10 Funding

This research was supported in part by the French National Spatial Agency (CNES) and the Midi-Pyrénées Region.

2.11 Supplemental Material

R_computation_details : Technical description about the computation in R of all presented methods on both datasets. (.pdf file)

university_of_pavia_x : *University of Pavia* spectral dataset used in the article. (.txt file) **(Too heavy. Only available on demand.)**

university_of_pavia_y : *University of Pavia* classes numbers. (.txt file)

wave_uni : *University of Pavia* wavelength. (.txt file)

Index_random_pavia_size30 : Lines index of the **university_of_pavia_x** matrix for 50 repetitions of 30 random learning samples per class. (.R file)

Pavia_test_R_codes : Application of all presented methods on the *University of Pavia* dataset for 50 repetitions of 30 random learning samples per class. (.R file)

Multifunc_classif : R-code to perform the Functional Multinomial Logistic method described in the article. (.R file)

Multifunc_help : Help file explaining the use of the Functional Multinomial Logistic method. (.pdf file)

Funopadi_classif : R-code to perform the NonParametric Functional Data Analysis method described in the article. (.R file)

Funopadi_help : Help file explaining the use of the NonParametric Functional Data Analysis method. (.pdf file)

Chapitre 3

Méthodes de sélection de variables spectrales dans un cadre prédictif

L'étude de données hyperspectrales conduit à l'exploration de données fortement corrélées dont l'information utile à la prédiction qu'elles contiennent est potentiellement redondante. De plus, la petite taille des échantillons face au nombre conséquent de variables dû à la grande finesse de discrétisation des données provoque l'apparition du «fléau de la dimension». La sélection de variables spectrales a été abordée d'un point de vue multivarié (non-fonctionnel) dans un cadre prédictif (régression ou classification supervisée). L'objectif de cette approche est de sélectionner un nombre restreint de bandes spectrales afin d'obtenir un modèle parcimonieux conservant une bonne capacité de prédiction. L'un des avantages de ce type de méthodes sélectives est qu'il propose des modèles plus interprétables que les modèles non-sélectifs.

La première partie de ce chapitre est consacrée à l'évaluation de la pertinence de deux méthodes parcimonieuses de sélection de variables dans un cadre de régression de données hyperspectrales : la méthode linéaire LASSO [202] et la méthode non-linéaire appelée «Most Predictive Design Points» (MPDP) [82] (partie issue de l'article [220]). La méthode LASSO est fondée sur un principe de minimisation d'un problème de moindres carrés pénalisés par la norme L^1 du vecteur des paramètres. La méthode MPDP combine un algorithme «pas à pas» de type forward avec un outil d'estimation non-paramétrique appelé régression linéaire locale. Afin de souligner l'intérêt de construire des modèles parcimonieux, ces deux méthodes ont également été comparées avec la méthode de régression Ridge [108], une méthode standard non sélective basée sur une régularisation de type L^2 des paramètres du modèle. Ces trois méthodes ont été appliquées sur des données provenant de relevés spectrométriques réalisés sur 25 prairies. La faible quantité de données disponible nous a contraint à nous placer dans un cadre d'échantillons de petite taille, du fait du nombre très important de bandes spectrales (typiquement, quelques dizaines de spectres pour plusieurs milliers de bandes spectrales). Ce travail permet de souligner (pour les données étudiées) la pertinence de réduire la dimension spectrale en sélectionnant les longueurs d'onde les plus prédictives. Cependant, l'instabilité des solutions obtenues, liée aux fortes corrélations présentes dans ce type de données, implique la nécessité de mener d'autres études plus approfondies sur le sujet.

La deuxième partie présente un algorithme de sélection de variables appelé «Nonlinear Parsimonious Feature Selection» (NPFS) pour la classification d'images hyperspectrales, construit à partir d'un classifieur basé sur les Modèles de Mélanges Gaussiens (partie issue de l'article [76]). Cet algorithme sélectionne de manière itérative les bandes spectrales maximisant une estimation du taux de bonne classification des données, et s'arrête lorsque l'amélioration apportée par l'ajout de variables n'est plus significatif ou lorsque le nombre maximal de variables est atteint. Cet algorithme opère une mise à jour du modèle par une réestimation des paramètres des

classes sans que la réestimation du modèle complet ne soit nécessaire. Cette approche permet également un accès direct aux sous-modèles par l'intermédiaire d'une marginalisation de la distribution Gaussienne. L'application de cet algorithme à deux jeux de données hyperspectraux a été comparée avec divers classifieurs de type Séparateurs à Vaste Marge (linéaire, à noyau polynomial ou gaussien, non-linéaire à élimination récursive de caractéristiques). Par rapport aux SVM, la méthode proposée sélectionne plus rapidement un faible nombre de caractéristiques informatives tout en conservant un potentiel de prédiction similaire et en facilitant l'interprétation des bandes spectrales extraites. Cette approche permet d'obtenir des taux comparables de bonne classification avec seulement 10% des variables spectrales initiales.

Sélection de variables pour l'imagerie hyperspectrale

Anthony Zullo[1,2] & Mathieu Fauvel[1] & Frédéric Ferraty[2]

[1] *Laboratoire DYNAFOR - UMR 1201 - INRA & INP Toulouse, Avenue de l'Agrobiopole, 31326 Castanet-Tolosan, France*
[2] *Institut de Mathématiques de Toulouse - UMR 5219 & Université de Toulouse, 118 route de Narbonne, 31062 Toulouse, France*
Adresses mail : anthony.zullo@toulouse.inra.fr ; mathieu.fauvel@ensat.fr ; ferraty@math.univ-toulouse.fr

Résumé. L'imagerie hyperspectrale est un domaine qui s'est développé récemment et qui nécessite le développement de nouvelles méthodes statistiques spécifiquement adaptées. Le principal problème engendré par ces images provient de la finesse de leur résolution spectrale, générant ainsi des données de grandes dimensions, et entraînant en conséquence l'apparition du problème statistique appelé «fléau de la dimension» se référant à la situation où le rapport nombre de variables sur taille d'échantillon est très grand. L'objectif de cette présentation est d'évaluer la pertinence de deux méthodes parcimonieuses, l'une linéaire et l'autre non linéaire, permettant de prédire une réponse scalaire à partir d'un petit nombre de variables explicatives. Nous nous focalisons sur la mise en œuvre de deux techniques statistiques dites «sélectives» dont l'objectif principal est de retenir un nombre raisonnable de variables explicatives tout en conservant un bon pouvoir prédictif. L'avantage de ce type de méthodes sélectives est qu'il propose des modèles plus interprétables. La première méthode sélective implémentée, appelée Lasso, permet de retenir les variables les plus explicatives dans le cadre d'un modèle de régression linéaire. La seconde est une méthode sélective non-paramétrique développée récemment qui combine un algorithme «pas à pas» de type forward avec un outil d'estimation non-paramétrique appelé régression linéaire locale. L'aspect non-paramétrique de cette méthode autorise la prise en compte de relations non linéaires. Les comportements de ces deux méthodes sont comparés sur un jeu de données hyperspectral selon un critère de validation croisée.

Mots-clés. Fléau de la dimension, imagerie hyperspectrale, Lasso, régression linéaire locale, sélection non-paramétrique de variables, validation croisée.

Abstract. Hyperspectral imaging is an area that has been developed recently and requires the development of new statistical methods specifically adapted. The main problem caused by these images comes from their fine spectral resolution, thereby generating high-dimensional data, and therefore causing the appearance of the statistical problem called "curse of dimensionality" referring to the situation where the relative number of variables on sample size is very large. The objective of this presentation is to assess the relevance of two sparse methods, one linear and one nonlinear, for the prediction of a scalar response from a small number of explanatory variables. We focus on the implementation of two statistical "selective" techniques whose main objective is to keep a reasonable number of variables while maintaining a good predictive power. The advantage of selective methods is that it provides more interpretable models. The first implemented selective method, called Lasso, keeps the most explanatory variables in the context of a linear regression model. The second is a more recently developed nonparametric method that combines a selective step-by-step forward type algorithm with a non-parametric estimation tool called local linear regression. The nonparametric aspect of this method allows the inclusion of nonlinear relations. Behavior in practice of these two methods are compared on a set of hyperspectral data using a cross-validation criterion.

Keywords. Curse of dimensionality, hyperspectral imaging, Lasso, local linear regression, non-parametric variable selection, cross-validation.

3.1 Introduction

L'étude d'images hyperspectrales a fait l'objet d'une attention particulière au cours des dix dernières années. Nous nous intéressons à des images hyperspectrales pour lesquelles, à chaque pixel i est associé, d'une part, un hyperspectre, courbe finement échantillonnée selon d longueurs d'onde $\lambda^1, \ldots, \lambda^d$, représenté par un vecteur aléatoire $X_i = (X_i^1, \ldots, X_i^d)$ de dimension d (i.e., $X_i^j = X_i(\lambda^j)$, $\forall j \in \{1, \ldots, d\}$), et d'autre part, une variable réponse quantitative Y_i. Étant donné $\{(X_i, Y_i),\ \forall i \in \{1, \ldots, n\}\}$ un échantillon d'apprentissage, le problème de «régression» consiste à associer à chaque pixel une estimation de la valeur de la variable réponse correspondante. La régression supervisée dans le cadre de l'étude d'images hyperspectrales est cependant une tâche difficile : la majorité des méthodes de régression n'est pas appropriée au traitement de telles données [121]. Le manque d'efficience de ces méthodes est principalement dû au concours de divers paramètres. En effet, les hyperspectres échantillonnés ont un nombre important d de bandes spectrales ; de plus, on se place dans le cas d'une petite taille n d'échantillon d'apprentissage (i.e., un petit nombre de pixels). D'un point de vue statistique, cela revient donc à considérer un jeu de données contenant un grand nombre de variables d pour un échantillon de petite taille n. Cette situation inconfortable est plus connue sous le nom de «fléau de la dimension» [62].
Parmi toutes les méthodes statistiques existantes, la sélection de variables regroupe un ensemble de méthodes permettant de résoudre des problèmes en grande dimension, notamment lorsque le nombre de variables est largement supérieur au nombre d'individus. Dans une telle configuration, on choisit en général d'émettre l'hypothèse que seules quelques variables (en nombre inférieur au nombre d'individus) suffisent à la construction d'un modèle permettant une résolution convenable du problème posé. On obtient ainsi un modèle plus interprétable qui peut même dans certains cas être «meilleur» (au sens d'un critère défini) que le modèle complet. Nous avons choisi de présenter une méthode non-linéaire de sélection de variables appelée *Most-Predictive Design Points* (MPDP) [82], et de comparer ses performances sur un jeu de données avec une autre méthode de sélection plus standard car linéaire appelée Lasso [202].

Dans la suite de cet article, nous présenterons ces deux méthodes avant de les appliquer pour les comparer sur un jeu de données particulier, puis nous conclurons quant aux résultats obtenus. Une troisième méthode non sélective appelée régression Ridge [108], sera aussi implémentée afin de souligner l'intérêt de construire des modèles parcimonieux.

3.2 Méthodologie statistique

Présentons brièvement les deux méthodes de sélection de variables qui seront appliquées sur notre jeu de données hyperspectral : la méthode Lasso, qui modélise de façon parcimonieuse et linéaire la relation entre la variable réponse et les variables explicatives, ainsi que la méthode MPDP, récemment développée, qui sélectionne les variables de façon non-paramétrique.

3.2.1 La méthode Lasso

Cette méthode est fondée sur le principe de minimisation d'un problème de moindres carrés pénalisés : $\hat{\beta}^L := \arg\min_{\beta} \{\sum_{i=1}^{n}(Y_i - \beta_0 - \sum_{j=1}^{d} X_i^j \beta_j)^2 + \lambda^L \sum_{j=1}^{d} |\beta_j|\}$, où λ^L est un paramètre de l'estimateur à régler. L'introduction d'une pénalité fondée sur la norme L^1 du vecteur des paramètres permet d'annuler un grand nombre de paramètres, ce qui revient à sélectionner un petit nombre de variables. Concernant la procédure d'estimation des paramètres, elle nécessite

l'utilisation d'une variante de l'algorithme appelé *Least Angle Regression* (LAR) [64] cherchant les variables explicatives X_i les plus linéairement corrélées avec les résidus successifs de la variable réponse Y. En pratique, le réglage du paramètre λ^L est remplacé par celui d'un paramètre de saturation du modèle s compris entre 0 (correspondant à la nullité de tous les coefficients estimés) et 1 (correspondant à l'estimation des moindres carrés).

3.2.2 La méthode *Most-Predictive Design Points* (MPDP)

Il s'agit d'une méthode statistique s'appuyant sur une méthode de régression non-paramétrique particulière : la régression linéaire locale [69]. L'algorithme permettant de réaliser cette sélection de variables est basé sur une procédure de type *Forward* pour sélectionner les variables les plus significatives. Cette méthode cherche le sous-ensemble de variables $\{X^{j_1}, \ldots, X^{j_p}\} \subset \{X^1, \ldots, X^d\}$ minimisant le critère de validation croisée de type LOOCV défini par $cv(X^{j_1}, \ldots, X^{j_p}) := \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{g}_{-i}(X_i^{j_1}, \ldots, X_i^{j_p}))^2$, où $\hat{g}_{-i}$ est l'estimateur de la régression linéaire locale calculé sans le $i^{ième}$ individu. Cet algorithme procède pas à pas : on cherche la variable la plus prédictive, puis parmi les variables restantes, on en déduit le couple le plus prédictif en conservant la première variable sélectionnée, et ainsi de suite jusqu'à atteindre un critère d'arrêt convenablement choisi. En pratique, la régression linéaire locale nécessite le choix d'un paramètre de lissage h obtenu par une validation croisée LOOCV.

3.3 Application aux données et comparaison des résultats

Le jeu de données étudié provient de relevés réalisés sur 25 prairies à l'aide d'un spectromètre pour des longueurs d'onde λ comprises entre 350 et 2400 nanomètres. Pour chaque prairie, on dispose d'une centaine d'hyperspectres environ. Ces données présentent la particularité d'être découpées en trois parties disjointes : 350-1350 nm, 1450-1800 nm et 2050-2400 nm, pour un total de 1698 variables explicatives. Ce découpage s'explique par une absorption athmosphérique des ondes sur les deux zones 1350-1450 nm et 1800-2050 nm. La variable réponse que l'on cherche à prédire, notée NV, représente le taux d'azote contenu dans chacune de ces prairies. Pour chacune des méthodes présentées précédemment, on obtient un modèle de la forme $Y_i = \mathcal{M}(X_i) + \varepsilon_i$, avec ε_i l'erreur associée au modèle, où $\mathcal{M}$ sélectionne les variables les plus pertinentes. Ces modèles sont ensuite comparés en utilisant le critère LOOCV relativement à la variance de la variable réponse : $LOOCVR(\mathcal{M}) = \dfrac{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \mathcal{M}_{-i}(X_i))^2}{var(Y)}$, où $\mathcal{M}_{-i}$ est le modèle construit à partir de l'échantillon d'apprentissage auquel on a enlevé le $i^{ième}$ individu. Pour des raisons de temps de calcul, nous nous sommes focalisés sur un échantillon d'apprentissage de taille modeste (250 hyperspectres, soit 10 hyperspectres aléatoirement sélectionnés dans chacune des 25 prairies). Concernant le réglage en pratique des paramètres de chacune des méthodes comparées, une sélection automatique a été réalisée pour le choix des paramètres s (équivalent à λ^L) pour la méthode Lasso et h pour la méthode MPDP.

TABLE 3.1 détaille les résultats obtenus pour cet échantillon ; la colonne intitulée «Ridge» correspond à la mise en œuvre de la méthode de régression Ridge sur ce même échantillon, servant de référence afin de comparer les méthodes de sélection de variables avec une méthode statistique standard non sélective. On constate que la méthode MPDP donne de meilleurs ré-

Modèle	Ridge	Lasso	MPDP
Valeur choisie pour le paramètre	$\lambda^R = 10$	$s = 0,144$	$h = 2,193$
Nombre de variables sélectionnées	1698	83	7
LOOCVR	$0,46$	$0,41$	$0,29$

TABLE 3.1 – Comparaison des méthodes de sélection de variables Ridge, Lasso et MPDP sur le jeu de données «prairies»

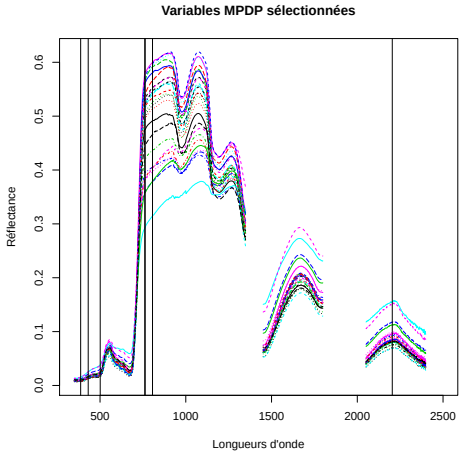


FIGURE 3.1 – Hyperspectres moyens des 25 prairies ; les lignes verticales localisent les longueurs d'onde correspondant aux variables sélectionnées par la méthode MPDP

sultats comparativement aux deux autres méthodes, tant sur le nombre de variables que sur la valeur du critère de validation croisée relative LOOCVR. En effet, alors que la régression Ridge conserve l'ensemble des variables et la méthode Lasso sélectionne 83 variables, la méthode MPDP sélectionne seulement 7 variables pour une valeur du critère LOOCVR inférieure aux deux autres.

FIGURE 3.1 représente les hyperspectres moyens des 25 prairies ainsi que les variables sélectionnées par la méthode MPDP. On constate que certaines variables sélectionnées sont situées dans une zone spectrale où les hyperspectres sont nettement distincts, contrairement aux autres, localisées dans des zones où les courbes se différencient difficilement.

3.4 Conclusion

Ces premiers résultats indiquent clairement la pertinence des méthodes sélectives dans le contexte de l'imagerie hyperspectrale. La méthode non-paramétrique MPDP améliore sensiblement les résultats obtenus par la méthode linéaire Lasso par l'obtention d'un modèle plus parcimonieux et possédant un pouvoir prédictif plus important. Cette étude est cependant incomplète ; des comparaisons plus fines devront être menées pour explorer la stabilité des résultats obtenus notamment en répétant plusieurs fois la construction d'échantillons-test et d'apprentissage.

Fast forward feature selection of hyperspectral images for classification with Gaussian mixture models

Mathieu Fauvel, Clément Dechesne, Anthony Zullo and Frédéric Ferraty

M. Fauvel, C. Deschene and A. Zullo are with the Université de Toulouse, INP-ENSAT, UMR 1201 DYNAFOR, France and with the INRA, UMR 1201 DYNAFOR, France

A. Zullo and F. Ferraty are with Institut de Mathématiques de Toulouse - UMR 5219 IMT & Université de Toulouse, France

Abstract. A fast forward feature selection algorithm is presented in this paper. It is based on a Gaussian mixture model (GMM) classifier. GMM are used for classifying hyperspectral images. The algorithm selects iteratively spectral features that maximizes an estimation of the classification rate. The estimation is done using the k-fold cross validation. In order to perform fast in terms of computing time, an efficient implementation is proposed. First, the GMM can be updated when the estimation of the classification rate is computed, rather than re-estimate the full model. Secondly, using marginalization of the GMM, sub models can be directly obtained from the full model learned with all the spectral features. Experimental results for two real hyperspectral data sets show that the method performs very well in terms of classification accuracy and processing time. Furthermore, the extracted model contains very few spectral channels.

Keywords. Hyperspectral image classification, nonlinear feature selection, Gaussian mixture model, parsimony.

3.5 Introduction

Since the pioneer paper of J. Jimenez and D. Landgrebe [121], it is well known that hyperspectral images need specific processing techniques because conventional ones made for multispectral/panchromatic images do not adapt well to hyperspectral images. Generally speaking, the increasing number of spectral channels poses theoretical and practical problems [62]. In particular, for the purpose of pixel classification, the spectral dimension needs to be handled carefully because of the "Hughes phenomenon" [115] : with a limited training set, beyond a certain number of spectral features, a reliable estimation of the model parameters is not possible.

Many works have been published since the 2000s to address the problem of classifying hyperspectral images. A non-exhaustive list should include techniques from the machine learning theory (Support Vector Machines, Random Forest, neural networks) [77], statistical models [121] and dimension reduction [26]. SVM, and kernel methods in general, have shown remarkable performances on hyperspectral data in terms of classification accuracy [30]. However, these methods may suffer from a high computational load and the interpretation of the model is usually not trivial.

In parallel to the emergence of kernel methods, the reduction of the spectral dimension has received a lot of attention. According to the absence or presence of training set, the dimension reduction can be unsupervised or supervised. The former tries to describe the data with a lower number of features that minimize a reconstruction error measure, while the latter tries to extract features that maximize the separability of the classes. One of the most used unsupervised feature extraction method is the principal component analysis (PCA) [121]. But it has been demonstrated that PCA is not optimal for the purpose of classification [47]. Supervised methods,

such as the Fisher discriminant analysis or the non-weighted feature extraction, have shown to perform better for the purpose of classification. Other feature extraction techniques, such as independent component analysis [211], have been applied successfully and demonstrate that even SVM can benefits from feature reduction [74, 73]. However, conventional supervised techniques suffer from similar problems than classification algorithms in high dimensional space.

Rather than supervised and unsupervised techniques, one can also distinguish dimension reduction techniques into *feature extraction* and *feature selection*. Feature extraction returns a linear/nonlinear combination of the original features, while feature selection returns a subset of the original features. While feature extraction and feature selection both reduce the dimensionality of the data, the latter is much more interpretable for the end-user. The extracted subset corresponds to the most important features for the classification, i.e., the most important wavelengths. For some applications, these spectral channels can be used to infer mineralogical and chemical properties [145].

Feature selection techniques generally need a criterion, that evaluates how the model built with a given subset of features performs, and an optimization procedure that tries to find the subset of features that maximizes/minimizes the criterion [185]. Several methods have been proposed according to that setting. For instance, an entropy measure and a genetic algorithm have been proposed in [41, Chapter 9], but the band selection was done independently of the classifier, i.e., the criterion was not directly related to the classification accuracy. Jeffries Matusita (JM) distance and steepest-ascent like algorithms were proposed in [184]. The method starts with a conventional sequential forward selection algorithm, then the obtained set of features is updated using local search. The method has been extended in [24] where a multiobjective criterion was used to take into account the class separability and the spatial variability of the features. JM distance and exhaustive search as well as some refinement techniques have been proposed also in [185]. However rather than extracting spectral features, the algorithm returns the average over a certain bandwidth of contiguous channels, which can make the interpretation difficult and often leads to select a large part of the electromagnetic spectrum. Similarly, spectral intervals selection was proposed in [120], where the criterion used was the square representation error (square error between the approximate spectra and the original spectra) and the optimization problem was solved using dynamic programming. These two methods reduce the dimensionality of the data, but cannot be used to extract spectral variables. Recently, forward selection and genetic algorithm driven by the classification error minimization have been proposed in [23].

Feature selection has been also proposed for kernel methods. A recursive scheme used to remove features that exhibit few influence on the decision function of a nonlinear SVM was discussed in [206]. Alternatively, a shrinkage method based on ℓ_1-norm and linear SVM has been investigated by Tuia *et al.* [207]. The authors proposed a method where the features are extracted during the training process. However, to make the method tractable in terms of computational load, a linear model is used for the classification, which can limit the discriminating power of the classifier. In [32], a dependence measure between spectral features and thematic classes is proposed using kernel evaluation. The measure has the advantage to be applicable to multiclass problem making the interpretation of the extracted features easier.

Feature selection usually provides good results in terms of classification accuracy. However, several drawbacks can be identified from the above mentioned literature :

— It can be very time consuming, in particular when nonlinear classification models are used.

— When linear models are used for the selection of features, performances in terms of classification accuracy are not satisfying and therefore another nonlinear classifier should be used after the feature extraction.

— For multiclass problem, it is sometimes difficult to interpret the extracted features when a collection of binary classifiers is used (e.g., SVM).

In this work, it is proposed to use a forward strategy, based on [82], that uses an efficient implementation scheme and allows to process a large amount of data, both in terms of number of samples and variables. The method, called *nonlinear parsimonious feature selection* (NPFS), selects iteratively a spectral feature from the original set of features and adds it to a pool of selected features. This pool is used to learn a Gaussian mixture model (GMM) and each feature is selected according to a classification rate. The iteration stops when the increased in terms of classification rate is lower than a user defined threshold or when the maximum number of features is reached. In comparison to other feature extraction algorithms, the main contributions of NPFS is the ability to select spectral features through a nonlinear classification model and its high computational efficiency. Furthermore, NPFS usually extracts a very few number of features (lower than 5 % of the original number of spectral features).

The remaining of the paper is organized as follows. Section 3.6 presents the algorithm with the Gaussian mixture model and the efficient implementation. Experimental results on three hyperspectral data sets are presented and discussed in Section 3.7. Conclusions and perspectives conclude the paper in Section 3.8.

3.6 Non linear parsimonious feature selection

The following notations are used in the remaining of the paper. $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$ denotes the set of training pixels, where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional pixel vector, $y_i = 1, \ldots, C$ is its corresponding class, C the number of classes, n the total number of training pixels and n_c the number of training pixels in class c.

3.6.1 Gaussian mixture model

For a Gaussian mixture model, it is supposed that the observed pixel is a realization of a d-dimensional random vector such as

$$p(\mathbf{x}) = \sum_{c=1}^{C} \pi_c p(\mathbf{x}|c), \tag{3.1}$$

where π_c is the proportion of class c ($0 \leq \pi_c \leq 1$ and $\sum_{c=1}^{C} \pi_c = 1$) and $p(\mathbf{x}|c)$ is a d-dimensional Gaussian distribution, i.e.,

$$p(\mathbf{x}|c) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}_c|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_c)^\top \mathbf{\Sigma}_c^{-1}(\mathbf{x} - \boldsymbol{\mu}_c)\right).$$

with $\boldsymbol{\mu}_c$ being the mean vector of class c, $\mathbf{\Sigma}_c$ being the covariance matrix of class c and $|\mathbf{\Sigma}_c|$ its determinant. Following the maximum *a posteriori* rule, a given pixel is classified to the class c if $p(c|\mathbf{x}) \geq p(k|\mathbf{x})$ for all $k = 1, \ldots, C$. Using the Bayes formula, the posterior probability can be written as

$$p(c|\mathbf{x}) = \frac{\pi_c p(\mathbf{x}|c)}{\sum_{k=1}^{C} \pi_k p(\mathbf{x}|k)}. \tag{3.2}$$

Therefore, the maximum *a posteriori* rule can be written as

$$\mathbf{x} \text{ belongs to } c \Leftrightarrow c = \arg\max_{k=1,\ldots,C} \pi_k p(\mathbf{x}|k). \tag{3.3}$$

By taking the log of eq. (3.3) the final decision function is obtained (also known as quadratic discriminant function)

$$Q_c(\mathbf{x}) = -(\mathbf{x} - \boldsymbol{\mu}_c)^\top \mathbf{\Sigma}_c^{-1}(\mathbf{x} - \boldsymbol{\mu}_c) - \ln(|\mathbf{\Sigma}_c|) + 2\ln(\pi_c). \tag{3.4}$$

Using standard maximization of the log-likelihood, the estimator of the model parameters are given by

$$\hat{\pi}_c = \frac{n_c}{n}, \tag{3.5}$$

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{n_c}\sum_{i=1}^{n_c}\mathbf{x}_i, \tag{3.6}$$

$$\hat{\boldsymbol{\Sigma}}_c = \frac{1}{n_c}\sum_{i=1}^{n_c}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)^\top. \tag{3.7}$$

with n_c is the number of sample of class c.

For GMM, the "Hughes phenomenon" is related to the estimation of the covariance matrix. If the number of training samples is not sufficient for a good estimation the computation of the inverse and of the determinant in eq.(3.4) will be very numerically unstable, leading to poor classification accuracy. For instance for the covariance matrix, the number of parameters to estimate is equal to $d(d+1)/2$: if $d = 100$ then 5050 parameters have to be estimated then the minimum number of training samples for the considered class should be at least 5050. Note in that case the estimation will be possible but not accurate. Feature selection tackles this problem by allowing the construction of GMM with a reduced number p of variables, with $p << d$ and $p(p+1)/2 < n_c$.

3.6.2 Forward feature selection

The forward feature selection works as follow [107, Chapter 3]. It starts with an empty pool of selected features. At each step, the feature that most improves an estimation of the classification rate is added to the pool. The algorithm stops either if the increase of the estimated classification rate is too low or if the maximum number of features is reached.

The k-fold cross-validation (k-CV) is used in this work to estimate the classification rate. To compute the k-CV, a subset is removed from $\mathcal{S}$ and the GMM is learned with the remaining training samples. A test error is computed with the removed training samples used as validation samples. The process is iterated k times and the estimated classification rate is computed as the mean test error over the k subsets of $\mathcal{S}$.

The efficient implementation of the NPFS relies on a fast estimation of the parameters of the GMM when the k-CV is computed. In the following, it will be shown that by using update rules of the parameters and the marginalization properties of the Gaussian distribution, it is possible to perform k-CV and forward selection quickly. As a consequence, the GMM model is learned only one time during the whole training step. The algorithm 1 presents a pseudo code of the proposed method.

3.6.3 Fast estimation of the model on $\mathcal{S}^{n-\nu}$

In this section, it is shown that each parameter can be easily updated when a subset is taken off $\mathcal{S}$.

Proposition 1 (Proportion) *The update rule for the proportion is*

$$\hat{\pi}_c^{n-\nu} = \frac{n\hat{\pi}_c^n - \nu_c}{n-\nu} \tag{3.8}$$

where $\hat{\pi}_c^{n-\nu}$ and $\hat{\pi}_c^n$ are the proportions of class c computed over $n-\nu$ and n training samples respectively, ν is the number of removed samples from $\mathcal{S}$, ν_c is the number of removed samples from class c such as $\sum_{c=1}^{C}\nu_c = \nu$.

Algorithm 1 NPFS pseudo code

Require: $\mathcal{S}$, k, `delta`, `maxvariable`

Randomly cut $\mathcal{S}$ into k subsets such as $\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_k = \mathcal{S}$ and $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$
Learn the full GMM with $\mathcal{S}$
Initialize the set of selected variables φ_s to empty set ($|\varphi_s| = 0$) and available variables φ_a to the original set of variables ($|\varphi_a| = d$)
while $|\varphi_s| \leq$ `maxvariable` **do**
  for all $\mathcal{S}_u \subset \mathcal{S}$ **do**
    Update the model using eq. (3.8), (3.9) and (3.10) (or their loocv counterparts) according to $\mathcal{S}_u$
    for all $s \subset \varphi_a$ **do**
      Compute the classification rate on $\mathcal{S}_u$ for each set of variables $\varphi_s \cap s$ using the marginalization properties
    end for
  end for
  Average the classification rate over the k-fold
  if Improvement in terms of classification rate w.r.t. previous iteration is lower than `delta` **then**
    break
  else
    Add the variable s corresponding to the maximum classification rate to φ_s and remove it from φ_a
  end if
end while

Proposition 2 (Mean vector) *The update rule for the mean vector is*

$$\hat{\boldsymbol{\mu}}_c^{n_c-\nu_c} = \frac{n_c \hat{\boldsymbol{\mu}}_c^{n_c} - \nu_c \hat{\boldsymbol{\mu}}_c^{\nu_c}}{n_c - \nu_c} \tag{3.9}$$

where $\hat{\boldsymbol{\mu}}_c^{n_c}$ and $\hat{\boldsymbol{\mu}}_c^{n_c-\nu_c}$ are the mean vectors of class c computed over the n_c and $n_c - \nu_c$ training samples respectively, $\hat{\boldsymbol{\mu}}_c^{\nu_c}$ is the mean vector of the ν_c removed samples from class c.

Proposition 3 (Covariance matrix) *The update rule for the covariance matrix is*

$$\begin{aligned} \hat{\boldsymbol{\Sigma}}_c^{n_c-\nu_c} &= \frac{n_c}{(n_c-\nu_c)} \hat{\boldsymbol{\Sigma}}_c^{n_c} - \frac{\nu_c}{(n_c-\nu_c)} \hat{\boldsymbol{\Sigma}}_c^{\nu_c} \\ &\quad - \frac{n_c \nu_c}{(n_c-\nu_c)^2} \left(\hat{\boldsymbol{\mu}}_c^{n_c} - \hat{\boldsymbol{\mu}}_c^{\nu_c}\right) \left(\hat{\boldsymbol{\mu}}_c^{n_c} - \hat{\boldsymbol{\mu}}_c^{\nu_c}\right)^\top \end{aligned} \tag{3.10}$$

where $\hat{\boldsymbol{\Sigma}}_c^{n_c}$ and $\hat{\boldsymbol{\Sigma}}_c^{n_c-\nu_c}$ are the covariance matrices of class c computed over the n_c and $n_c - \nu_c$ training samples respectively.

3.6.4 Particular case of leave-one-out cross-validation

When very few training samples are available, it is sometimes necessary to resort to leave-one-out cross-validation ($k = n$). Update rules are still valid, but it is also possible to get a fast update of the decision function. If the removed sample does not belong to class c, only the proportion term in eq. (3.4) changes, therefore the updated decision rule can be written as :

$$Q_c^{n_c-1}(\mathbf{x}_n) = Q_c^{n_c}(\mathbf{x}_n) + 2 \ln \left(\frac{n-1}{n}\right). \tag{3.11}$$

where $Q_c^{n_c}$ and $Q_c^{n_c-1}$ are the decision rules for class c computed with n_c and $n_c - 1$ samples respectively. If the removed sample $\mathbf{x}_n$ belongs to class c then updates rules become :

Proposition 4 (Proportion-loocv)

$$\hat{\pi}_c^{n-1} = \frac{n\hat{\pi}_c^n - 1}{n-1} \tag{3.12}$$

Proposition 5 (Mean vector-loocv)

$$\hat{\boldsymbol{\mu}}_c^{n_c-1} = \frac{n_c \hat{\boldsymbol{\mu}}_c^{n_c} - \mathbf{x}_n}{n_c - 1} \tag{3.13}$$

Proposition 6 (Covariance matrix-loocv)

$$\begin{aligned} \hat{\boldsymbol{\Sigma}}_c^{n_c-1} &= \frac{n_c}{n_c-1}\hat{\boldsymbol{\Sigma}}_c^{n_c} \\ &\quad -\frac{n_c}{(n_c-1)^2}\left(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_c^{n_c}\right)\left(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_c^{n_c}\right)^\top . \end{aligned} \tag{3.14}$$

where $n_c - 1$ denotes that the estimation is done with only $n_c - 1$ samples rather than the n_c samples of the class.

Update rules have been proposed in [109] for the leave-one-out case. Authors have proposed a way to compute the inverse of the covariance matrix with a low computational cost when one sample is removed. It is based on the Sherman-Morrison-Woodbury formula. In their approach, the inverse of the covariance matrix is computed explicitly in eq. (3.4). In this work, we choose to not compute the inverse but rather solve the linear problem $z = \boldsymbol{\Sigma}^{-1}\mathbf{x}$. This approach is more demanding in terms of processing time (still fast when the number of variables if low $\sim$ 10-15) but far more robust in terms of numerical stability. An update rule for the case where the sample belongs the class c can be written by using the Cholesky decomposition of the covariance matrix and rank-one downdate, but the downdate step is not numerically stable and not used here.

3.6.5 Marginalization of Gaussian distribution

To get the GMM model over a subset of the original set of features, it is only necessary to drop the non-selected features from the mean vector and the covariance matrix [174]. For instance, let $\mathbf{x} = [\mathbf{x}_s, \mathbf{x}_{ns}]$ where $\mathbf{x}_s$ and $\mathbf{x}_{ns}$ are the selected variables and the non-selected variables respectively, the mean vector can be written as

$$\hat{\boldsymbol{\mu}} = [\boldsymbol{\mu}_s, \boldsymbol{\mu}_{ns}]^\top \tag{3.15}$$

and the covariance matrix as

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{s,s} & \boldsymbol{\Sigma}_{s,ns} \\ \boldsymbol{\Sigma}_{ns,s} & \boldsymbol{\Sigma}_{ns,ns} \end{bmatrix}. \tag{3.16}$$

The marginalization over the non-selected variables shows that $\mathbf{x}_s$ follows also a Gaussian distribution with mean vector $\boldsymbol{\mu}_s$ and covariance matrix $\boldsymbol{\Sigma}_{s,s}$. Hence, once the full model is learned, all the sub-models built with a subset of the original variables are available at no computational cost.

3.7 Experimental results

3.7.1 Data

Two data sets have been used in the experiments. The first data set has been acquired in the region surrounding the volcano Hekla in Iceland by the AVIRIS sensor. 157 spectral channels from 400 to 1,840 nm were recorded. 12 classes have been defined for a total of 10,227 referenced pixels. The second data set has been acquired by the ROSIS sensor during a flight campaign over Pavia, nothern Italy. 103 spectral channels were recorded from 430 to 860 nm. 9 classes have been defined for a total of 42776 referenced pixels.

TABLE 3.2 – Classification accuracies for *Hekla* data set. The results correspond to the mean value and variance of the overall accuracy over the 50 repetitions. The best result for each training setup is reported in bold face. n-NPFS and 5-NPFS correspond to the NPFS computed with the leave-one-out and 5-fold cross-validation, respectively. RFE, SVM_{ℓ_1} and $\text{SVM}^p_{\ell_1}$ correspond to the recursive feature extraction SVM, the linear SVM with ℓ_1 constraint and the linear SVM with ℓ_1 with the explicit order 2 polynomial feature space, respectively. SVM_{poly} and $\text{SVM}_{\text{gauss}}$ correspond to the conventional nonlinear SVM with a order 2 polynomial kernel and Gaussian kernel, respectively.

n_c	n-NPFS	5-NPFS	RFE	SVM_{ℓ_1}	$\text{SVM}^p_{\ell_1}$	SVM_{poly}	$\text{SVM}_{\text{gauss}}$
50	**92.5 ± 1.2**	92.4 ± 1.2	90.2 ± 1.8	90.3 ± 1.0	91.6 ± 0.6	84.6 ± 1.6	90.4 ± 1.6
100	94.8 ± 0.7	94.6 ± 0.6	**95.6 ± 0.3**	93.9 ± 0.5	94.8 ± 0.1	91.4 ± 0.4	**95.6 ± 0.3**
200	95.9 ± 0.3	95.8 ± 0.3	**96.8 ± 1.1**	95.6 ± 0.1	96.3 ± 0.1	95.5 ± 0.1	**96.8 ± 1.1**

For each data set, 50, 100 and 200 training pixels per class were randomly selected and the remaining referenced pixels were used for the validation. 50 repetitions were done for which a new training set has been generated randomly.

3.7.2 Competitive methods

Several conventional feature selection methods have been used as baseline.

— Recursive Feature Elimination (RFE) for nonlinear SVM [206]. In the experiment, a Gaussian kernel was used.

— Linear SVM with ℓ_1 (SVM_{ℓ_1}) constraint on the feature vector [207] based on the LIBLINEAR implementation [70].

— To overcome the limitation of the linear model used in LIBLINEAR, an explicit computation of order 2 polynomial feature space has been used together with LIBLINEAR ($\text{SVM}^p_{\ell_1}$). Formally, a nonlinear transformation ϕ has been apply on the original samples :

$$\begin{aligned} \mathbb{R}^d &\rightarrow \mathbb{R}^p \\ \mathbf{x} = [x_1, \ldots, x_d] &\mapsto \phi(\mathbf{x}) = \Big[x_1, \ldots, x_d, x_1^2, x_1x_2, \ldots, \\ &\qquad x_1x_d, x_2^2, x_2x_3, \ldots, x_d^2\Big] \end{aligned}$$

with $p = \binom{2+d}{2}$. For *Hekla* data and *University of Pavia* data, the dimension p of the projected space is 12561 and 5460, respectively.

For comparison, a SVM with a Gaussian kernel and a order 2 polynomial kernel classifier, based on the LIBSVM [40], with all the variables have been used too.

For the linear/nonlinear SVM, the penalty parameter and the kernel hyperparameters were selected using 5-fold cross-validation. For NPFS, the threshold (`delta` in Algorithm 1) was set to 0.5% and the maximum number of extracted features was set to 20. The estimation of the error has been computed with a leave-one-out CV (n-NPFS) and a 5-fold CV (5-NPFS). Each variable has been standardized before the processing (i.e., zero mean and unit variance).

3.7.3 Results

The mean accuracies and the variance over the 50 runs are reported in Table 3.2 and Table 3.3. The mean numbers of extracted features for the different methods are reported in Figure 3.2 and Figure 3.3.

TABLE 3.3 – Classification accuracies for *University of Pavia* data set. Same notations as in Table 3.2.

n_c	n-NPFS	5-NPFS	RFE	SVM_{ℓ_1}	$\text{SVM}^p_{\ell_1}$	SVM_{poly}	$\text{SVM}_{\text{gauss}}$
50	82.2 ± 4.4	83.4 ± 7.6	84.7 ± 4.0	75.1 ± 2.5	81.0 ± 2.8	82.9 ± 3.4	**84.8 ± 3.4**
100	86.3 ± 3.2	85.9 ± 3.1	**88.4 ± 0.9**	77.3 ± 1.4	83.6 ± 1.3	86.5 ± 1.6	88.4 ± 1.4
200	87.7 ± 3.1	87.9 ± 1.9	**90.8 ± 0.3**	78.5 ± 0.7	85.5 ± 0.4	88.8 ± 0.6	**90.8 ± 0.3**

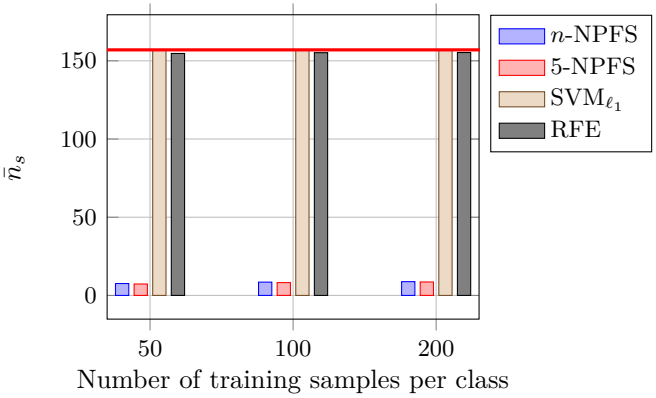


FIGURE 3.2 – Mean number $\bar{n}_s$ of selected features for the different methods for *Hekla* data set. The red line indicates the original number of spectral features. Projected ℓ_1 SVM is not reported because the mean number of extracted features was too high (e.g., 6531 for n_c=50).

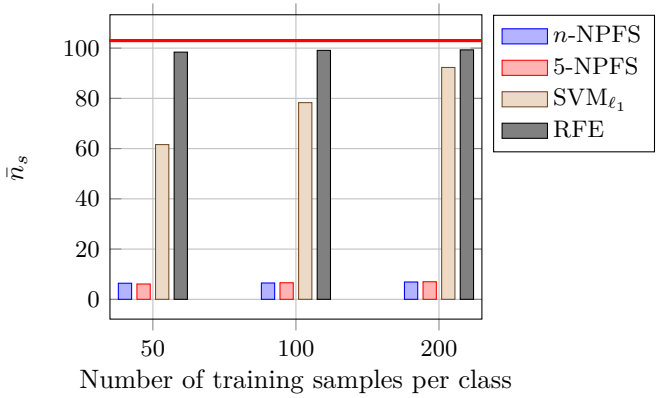


FIGURE 3.3 – Mean number $\bar{n}_s$ of selected features for the different methods for *University of Pavia* data set. The red line indicates the original number of spectral features. Projected ℓ_1 SVM is not reported because the mean number of extracted features was too high (e.g., 5110 for n_c=50).

From the tables, it can be seen that there is no difference in the results obtained with n-NPFS or 5-NPFS. They perform equally on both data sets in terms of classification accuracy or number of extracted features. However, 5-NPFS is much faster in terms of computation time.

RFE and $\text{SVM}_{\text{gauss}}$ provide the best results in terms of classification accuracy, except for the *Hekla* data set and $n_c = 50$. From the figure, it can be seen that the number of extracted features is almost equal to original number of spectral features, meaning that in these experiments RFE is equivalent to $\text{SVM}_{\text{gauss}}$. Hence, RFE was not able to extract few relevant spectral features.

ℓ_1 SVM applied on the original features or the projected features is not able to extract relevant features. In terms of classification accuracy, the linear SVM does not perform well for the *University of Pavia* data set. Nonlinear ℓ_1 SVM provides much better results for both data sets. In comparison to the non sparse nonlinear SVM computed with an order 2 polynomial kernel, ℓ_1 nonlinear SVM performs better in terms of classification accuracy for the *Hekla* data while it performs worst for the *University of Pavia* data.

In terms of number of extracted features, NPFS provides the best results, by far, with an average number of extracted features equal to 5% of the original number. All the other methods were not able to extract few features without decreasing drastically the overall accuracy. For instance, for the *Hekla* data set and $n_c = 50$, only 7 spectral features are used to build the GMM and lead to the best classification accuracy. A discussion on the extracted features is given in

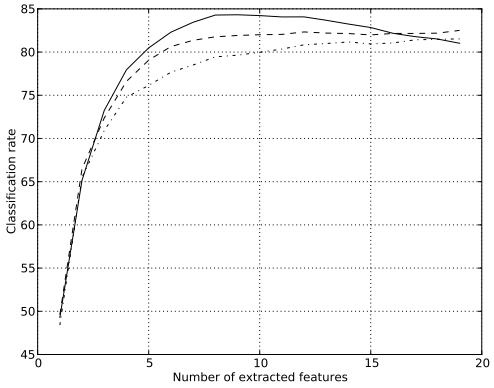


FIGURE 3.4 – Classification rate in function of the number of extracted features. Continuous line corresponds to 5-NPFS, dashed line to SVM with a Gaussian kernel and dash-doted line to a linear SVM.

TABLE 3.4 – Mean processing time in seconds in function of the number of samples per class for the *University of Pavia* data set. 20 repetitions have been done on laptop with 8Gb of RAM and Intel(R) Core(TM) i7-3667U CPU @ 2.00GHz processor.

n_s	50	100	200	400
$\text{SVM}_{\text{gauss}}$	11	40	140	505
SVM_{ℓ_1}	52	115	234	498
n-NPFS	242	310	472	883
5-NPFS	35	31	29	43

the next section.

Figure 3.4 presents the average classification rate of 5-NPFS, SVM with a Gaussian kernel and a linear SVM applied on the features selected with 5-NPFS. 20 repetitions have been done on the University data set with n_c=50. The optimal parameters for SVM and linear SVM have been selected using 5-fold cross-validation. From the figure, it can be seen that the three algorithms have similar trends. When the number of features is relatively low (here lower than 15) GMM performs the best, but when the number of features increases too much, SVM (non linear and linear) performs better in terms of classification accuracy. It is worth noting that such observations are coherent with the literature : SVM are known to perform well in high dimensional space, while GMM is more affected by the dimension.

The mean processing time for the *University of Pavia* data set for several training set sizes is reported in Table 3.4. It includes parameter optimization for $\text{SVM}_{\text{gauss}}$ and SVM_{ℓ_1}. Note that the RFE consists in several $\text{SVM}_{\text{gauss}}$ optimizations, one for each feature removed (hence, if 3 features are removed, the mean processing time is approximately multiplied by 3). It can be seen that the 5-NPFS method is little influenced by the size of the training set : what is important is the number of (extracted) variables. For $n_s = 50$, the processing time is slightly higher because of overload due to parallelization procedure. n-NPFS is the more demanding in terms of processing time and thus should be used only when the number of training samples is very limited. Finally, it is important to underline that the NPFS is implemented in Python while SVM used a state of the art implementation in C++ [40].

From these experiments, and from a practical viewpoint, NPFS is a good compromise between high classification accuracy and sparse modeling.

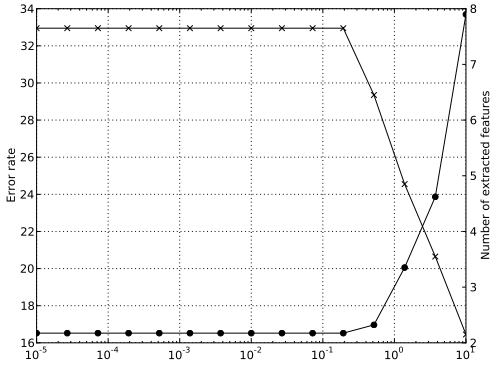


FIGURE 3.5 – The dotted line and the crossed line represent the mean error rate and the average number of selected features, respectively, as a function of delta. The simulation was done on the *University of Pavia data set, with* $n_c = 50$ *and for the 5-NPFS algorithm.*

3.7.4 Discussion

The extracted channels by 5-NPFS and n-NPFS were compared for one training set of the *University of Pavia* data set : two channels were the same for both methods, 780nm and 776nm ; two channels were very close, 555nm and 847nm for 5-NPFS and 551nm and 855nm for n-NPFS ; one channel was close, 521nm for 5-NPFS and 501nm for n-NPFS. The other channel selected with n-NPFS is 772nm. If the process is repeated, the result is terms of features selected with n-NPFS and 5-NPFS is similar : on average 35% of the selected features are identical (not necessarily the first ones) and the other selected features are close in terms of wavelength.

The influence of the parameter `delta` has been investigated on the *University of Pavia* data set. 20 repetitions have been done with $n_c = 50$ for several values of `delta`. Results are reported on Figure 3.5. From the figure, it can be seen that when delta is set to a value larger than approximately 1%, the algorithm stops too early and the number of selected features is too low to perform well. Conversely, setting `delta` to a small value does not change the classification rate, a plateau being reached for `delta` lower than 0.5%. In fact, because of the "Hughes phenomenon", adding spectral features to the GMM will first lead to an increase of the classification rate but then (after a possible plateau) the classification rate will decrease, i.e., the improvement after two iterations of the algorithm will be negative.

Figure 3.6 presents the most selected features for the *University of Pavia* data set. 1000 random repetitions have been done with n_c=200 and the features shaded in the figure have been selected at least 10% times (i.e., 100 times over 1000) using 5-NPFS. Five spectral domains can be identified, two from the visible range and three from the near infrared range. In particular, it can be seen that spectral channels from the red-edge part are selected. The width of the spectral domain indicates the variability of the selection. The high correlation between adjacent spectral bands makes the variable selection "unstable", e.g., for a given training set, the channel t would be selected but for another randomly selected training set it might be the channel $t+1$ or $t-1$. It is clearly a limitation of the proposed approach.

To conclude this discussion, similar spectral channels are extracted with n-NPFS and 5-NPFS, while the latter is much faster. Hence, n-NPFS should be only used when very limited number of samples is available. A certain variability is observed in the selection of the spectral channels due to the high correlation of adjacent spectral channels and the step-wise nature of the method.

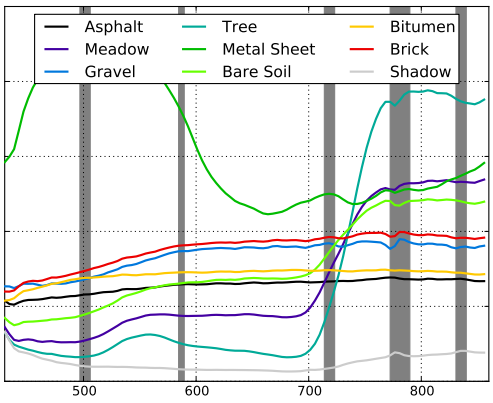


FIGURE 3.6 – Most selected spectral domain for the *University of Pavia* data set. Gray bars correspond to the most selected parts of the spectral domain. Horizontal axis corresponds to the wavelength (in nanometers). The mean value of each class is represented in continuous colored lines.

3.8 Conclusion

A nonlinear parsimonious feature selection algorithm for the classification of hyperspectral images has been presented. Using a Gaussian mixture model classifier, spectral variables are extracted iteratively based on the cross-validation estimate of the classification rate. An efficient implementation is proposed that takes into account some properties of Gaussian mixture model : a fast update of the model parameters and a fast access to the sub-models. Experimental results show that the proposed method is able to select few relevant features, and outperforms standard SVM-based sparse algorithms while reaching similar classification rates to those obtained with SVM. Furthermore, in comparison to SVM based feature selection algorithm, multiclass problem is handled by the GMM making the interpretation of the extracted channels easier.

More investigation are needed to fully understand which features are extracted, since the method is purely statistical. If the red-edge has been identified, the other extracted features are not clearly interpretable. Moreover, small variability has been observed due to the high correlation between adjacent bands and the step-wise procedure. To overcome this limitation, a continuous interval selection strategy, as in [185], will be investigated. Also, a steepest-ascent search strategy could be used to make the final solution more stable.

The python code of the algorithm is available freely for download : `https://github.com/mfauvel/FFFS`.

3.9 Acknowledgment

The authors would like to thank Professor P. Gamba, University of Pavia, for providing the *University of Pavia* data set and Professor J.A. Benediktsson, University of Iceland, for providing the *Hekla* data set. They would like also to thank the reviewers for their many helpful comments.

Chapitre 4

Modélisation du bruit dans les données hyperspectrales par un modèle hétéroscédastique

Dans ce chapitre, on cherche à appréhender le problème du bruit dans les données hyperspectrales. Ces données sont souvent perturbées de manière inégale et les sources de bruit sont multiples (bandes d'absorption dans l'atmosphère, bruit numérique dû au capteur, corrections géométriques, ...). Les données observées sont donc contaminées par un bruit dont le profil spectral est totalement inconnu. L'une des principales difficultés liées à cette situation réside dans le fait que le bruit dans les données est souvent hétéroscédastique, c'est-à-dire qu'il varie selon les longueurs d'ondes mais aussi selon les spectres. L'étude de ces données bruitées peut ainsi causer une dégradation plus ou moins importante de la capacité de prédiction de méthodes pourtant adaptées au traitement de données hyperspectrales. Afin de limiter l'impact du bruit dans les données, nous développons une approche associant méthode fonctionnelle et débruitage des données.

Ce travail propose une méthode de traitement de données hyperspectrales bruitées combinant un modèle non-paramétrique fonctionnel avec un lissage (i.e., débruitage) «adaptatif» des données (partie issue de l'article [88]). L'aspect «adaptatif» du lissage à noyau proposé est basé sur l'hypothèse selon laquelle l'intensité du bruit varie en fonction de la longueur d'onde : pour chaque bande spectrale considérée, plus la variabilité inter-spectres est élevée, plus le lissage appliqué sera important. L'estimateur non-paramétrique fonctionnel est alors construit à partir de variables fonctionnelles lissées au lieu des variables initiales bruitées. Il est important de remarquer que le lissage des données et l'estimation de l'opérateur de régression sont emboîtés de sorte que leurs paramètres respectifs optimisent un critère prédictif global. Ce nouvel estimateur non-paramétrique fonctionnel avec lissage imbriqué a été étudié d'un point de vue théorique, montrant une absence de dégradation vis-à-vis de la vitesse de convergence de l'estimateur non-paramétrique fonctionnel standard si l'on observait les données non bruitées, sous certaines hypothèses raisonnables dans le cadre de l'étude de données hyperspectrales. Dans la pratique, le paramètre de lissage a été choisi comme proportionnel à l'écart-type des données calculé en chaque bande spectrale de la discrétisation. L'approche non-paramétrique fonctionnelle avec lissage intégré a été appliquée à des données simulées ainsi qu'à deux jeux de données hyperspectrales de diverses natures. Cette méthode a été comparée avec sa version standard sans lissage sur des données simulées bruitées dans un cadre de régression. Quatre configurations de bruit (dont trois hétéroscedastiques) ont été étudiées : un bruit constant suivant les longueurs d'onde et les spectres, une croissance puis une décroissance exponentielle du bruit suivant les longueurs d'onde, un bruit dichotomique selon les longueurs d'onde, un bruit dicho-

tomique selon les longueurs d'onde avec une zone d'amplification variant selon les spectres. Ces deux estimateurs non-paramétriques fonctionnels (avec ou sans débruitage) ont été comparés sur ces simulations pour quatre niveaux de bruit différents. Dans un cadre de régression, les deux méthodes non-paramétriques fonctionnelles (avec ou sans débruitage) ont été comparées sur des données simulées par un modèle biophysique, pour deux fréquences d'échantillonnage différentes. Ces données, non bruitées de par leur conception, ont été perturbées à l'aide d'un modèle de bruit dichotomique entre le domaine visible et le proche infrarouge (modèle associé aux capteurs hyperspectraux) pour trois niveaux de bruit différents. La méthode non-paramétrique fonctionnelle avec lissage imbriqué et sa version standard sans lissage ont également été mises en compétition avec des méthodes multivariées pour la classification supervisée de données hyperspectrales réelles (données *MADONNA*) avec une contrainte de petite taille de l'échantillon attribué à la construction du modèle. Le principal intérêt de l'association de méthodes statistiques fonctionnelles et du débruitage des données réside dans l'amélioration très significative des résultats de prédictions qui en découlent. Ainsi, ce nouvel estimateur s'adapte aux profils spectraux bruités, même dans un cadre de bruit hétéroscédastique complexe. Par ailleurs, la méthode imbriquée proposée est d'autant moins impactée par l'intensité du bruit dans les données que le nombre de points de discrétisation est grand devant le nombre de pixels (ce qui est souvent le cas lors de traitements de données hyperspectrales). Cette méthode a notamment donné les meilleurs résultats sur des données hyperspectrales réelles supposées bruitées mais dont le véritable profil spectral est totalement inconnu.

Nonparametric regression on contaminated functional predictor with application to hyperspectral data

Frédéric Ferraty[1], Anthony Zullo[1,2], Mathieu Fauvel[2]

[1] Institut de Mathématiques de Toulouse - UMR 5219 & Université de Toulouse,
118 route de Narbonne, 31062 Toulouse, France
[2] Laboratoire DYNAFOR - UMR 1201 - INRA & INP Toulouse,
Avenue de l'Agrobiopole, 31326 Castanet-Tolosan, France

Abstract. We propose to regress nonparametrically a scalar response Y on a random curve X when only a contaminated version X^* of X is observable at some measurement grid. To override this common setting, a kernel presmoothing step is achieved on the noisy functional predictor X^*. Afterthen, the kernel estimator of the regression operator is built using the smoothed functional covariate instead of the original corrupted ones. Rate of convergence is stated for this nested-kernel estimator with a special attention on high-dimensional setting (i.e the size of the measurement grid is much larger than the sample size). The proposed method is applied on simulated datasets in order to assess its finite-sample properties. Our methodology is further illustrated on a real hyperspectral dataset involving a supervised classification problem.

Keywords. errors-in-variables ; functional data ; high-dimensional setting ; hyperspectral image ; nonparametric functional regression ; random curve ; supervised classification ; teledetection

4.1 Introduction

With technological progress, more and more sophisticated sensors or monitoring devices allow to observe a sample of curves in addition to other standard variables. Handling and modelling such a collection of curves is a standard problematic in Functional Data Analysis (FDA) and statisticians have developed methodologies especially designed to this situation (see for instance Ramsay and Silverman [173], Ferraty and Romain [84], Bosq [16], Horváth and Kokoszka [111] and Hsing and Eubank [113]).

Regressing nonparametrically a scalar response Y on a random curve X (i.e. functional variable taking its values in some real valued univariate function space) is a standard and popular methodology of Functional Data Analysis (FDA), especially when one wishes to relax as much as possible assumptions on the regression model (see for intance Ferraty and Vieu [87] and references therein). It corresponds to a model $Y = r(X) + error$ where the regression operator r satisfies only regularity-type conditions (continuity, Lipschitz, differentiability, etc). Given a sample $\{(X_i, Y_i); i = 1, \ldots, n\}$, as soon as we are able to build an estimator $\widehat{r}$ of r, one can easily derive a prediction $\widehat{Y}_{n+1}$ from a new curve X_{n+1} by the relationship $Y_{n+1} = \widehat{r}(X_{n+1})$. However, in numerous situations, instead of collecting directly X, one observes only a noisy version X^* of X (i.e. $X^* = X + noise$). To give an idea on this problematic, Figure 4.1 displays a sample of curves and corresponding contaminated profiles in various situations. It corresponds to the usual errors-in-variables topic but in functional covariate context when errors arise from imprecise measurement. In such a situation, it is clear that corrupted functional data may affect significantly the estimation of the regression operator and hence the prediction of the corresponding responses.

The general problem of errors-in-variables has been intensively studied in the non-functional nonparametric regression setting. The reader will find a useful overview in Carroll *et al.* [37] whereas more recent references focusing on nonparametric kernel methods are available in Delaigle [58]. When considering functional predictor, Yao *et al.* [217], Crambes *et al.* [53] and Wu

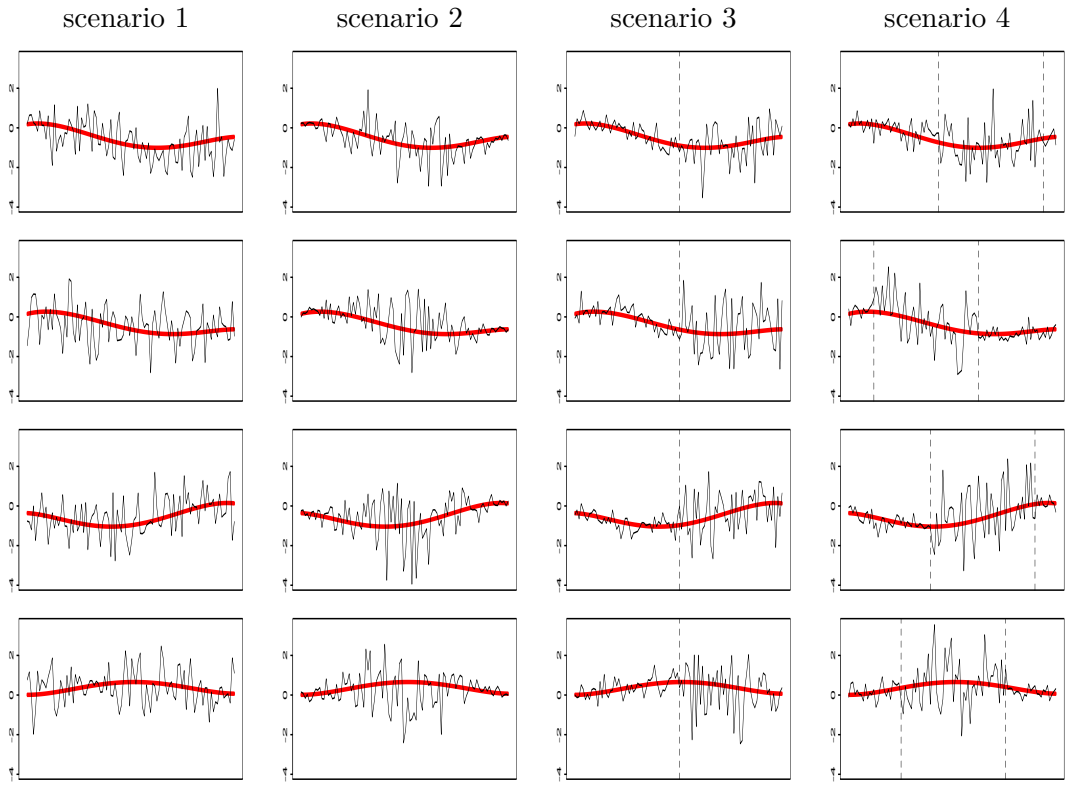


FIGURE 4.1 – For each contamination scenario, sample of curves with (thin lines) and without (thick lines) measurement errors : scenario 1 → errors with constant standard deviation, scenario 2-4 → errors with varying standard deviations (see Section 4.5.1 for more details).

et al. [215] developed estimating procedures able to handle measurement errors in the functional linear regression. More recently Radchenko *et al.* [171] proposed a new errors-in-variables approach in the functional single index.

Although errors-in-variables is a common setting for practitioners, it is surprising that to our current knowledge, there is no measurement errors approach in functional nonparametric regression model. So, this article proposes first developments to bridge the gap between pure nonparametric regression and functional predictor with measurement errors. To this end, a presmoothing step acting on the noisy functional covariate is achieved. Afterthen, a kernel estimator of the regression operator r, based on the denoised functional covariate, is built. As the presmoothing step involves a kernel smoother, the whole kernel estimator is named nested-kernel estimator. Despite the simplicity of the developped methodology, it is worth noting that this work provides the first theoretical results in such setting while proposing an implementation especially designed for practitioners dealing with hyperspectral datasets. It is especially emphasized that presmoothing the functional predictor does not impact the standard rate of convergence of the kernel estimator of the functional nonparametric regression operator as soon as the functional predictor is sampled at a frequency high enough (i.e. the number of measurements is much larger than the sample size). In addition, the implementation part of the nested-kernel estimator allows to illustrate the role played by the sampling frequency of the functional predictor in the predictive performances.

The remainder of this paper is organized as follows. Section 4.2 defines the nested-kernel estimator of the nonparametric regression operator and depicts the whole methodology for im-

plementing the estimating procedure. Section 4.3 explains how the methodology can be adapted to the supervised classification setting. Main theoretical results are given in Section 4.4. Section 4.5 is devoted to the implementation of the nested-kernel estimator. Finite sample properties are assessed by means of simulated data, including a simulated hyperspectral dataset coming from the teledetection community. To illustrate our methodology in a real situation, an hyperspectral image involving a supervised classification problem is used. Proofs are postponed in the appendix Section 4.7.

4.2 Estimating procedure and methodology

Let us first consider n pairs $(X_1, Y_1), \ldots, (X_n, Y_n)$ identically and independently distributed as (X, Y), where $X = \{X(\lambda);\, \lambda \in \Lambda\}$ ($\Lambda \subset \mathbb{R}$) is a random curve taking its values in L^2_Λ (i.e. space of squared integrable functions over Λ) and Y a real random variable (r.r.v.). We are interested in estimating nonparametrically the regression operator r such that $Y = r(X) + \varepsilon$ where ε is a zero-mean square-integrable r.r.v. independent of X. However, as indicated in the introduction, X is not observable ; only observations of a contaminated version X^* of X is available : for any $\lambda \in \Lambda$, $X^*(\lambda) = X(\lambda) + \eta(\lambda)$ where $\eta(\lambda)$ is the measurement error process valued at λ ; η is assumed to be a random process independent of (X, Y). So, our main task consists in building an estimator of r from $(X_1^*, Y_1), \ldots, (X_n^*, Y_n)$ instead of using $(X_1, Y_1), \ldots, (X_n, Y_n)$. In practice, the random curve $X^* = \{X^*(\lambda);\, \lambda \in \Lambda\}$ is not continuously observed over Λ but only sampled at a grid λ_0, λ_1, …, λ_d of size $d + 1$. The first step of our estimating procedure amounts to denoise the corrupted functional data by using a standard univariate kernel smoother. So, for any $\lambda \in \Lambda$ and $i = 1, \ldots, n$, the estimator $\widehat{X}_i(\lambda)$ of the unobservable quantity $X_i(\lambda)$ is obtained by means of the standard kernel smoother :

$$\widehat{X}_i(\lambda) = \frac{\sum_{j=0}^{d} X_i^*(\lambda_j)\, K_s\left\{h_s^{-1}(\lambda - \lambda_j)\right\}}{\sum_{j=0}^{d} K_s\left\{h_s^{-1}(\lambda - \lambda_j)\right\}}, \tag{4.1}$$

where $K_s(\cdot)$ is a given univariate symmetric kernel function and h_s a nonnegative smoothing parameter (i.e. bandwidth). Once the functional data is denoised, one can estimate nonparametrically the regression operator by using a functional kernel estimator. For any $x \in L^2_\Lambda$, the estimator $\widehat{r}$ of the regression operator r is defined from $(\widehat{X}_1, Y_1), \ldots, (\widehat{X}_n, Y_n)$:

$$\widehat{\widehat{r}}(x) = \frac{\sum_{i=1}^{n} Y_i\, K_r\left\{h_r^{-1}\|x - \widehat{X}_i\|_2\right\}}{\sum_{i=1}^{n} K_r\left\{h_r^{-1}\|x - \widehat{X}_i\|_2\right\}},$$

where $\|.\|_2$ stands for the standard norm in L^2_Λ, $K_r(\cdot)$ is an asymmetrical kernel function and h_r a bandwidth. The resulting estimator $\widehat{\widehat{r}}$ of r is clearly built from two embedded kernel estimators and this is why $\widehat{\widehat{r}}$ is called nested-kernel estimator and denoted with a double hat. The whole estimating procedure starting from $(X_1^*, Y_1), \ldots, (X_n^*, Y_n)$ up to the computation of $\widehat{\widehat{r}}$ amounts to select both smoothing parameters h_s and h_r. This optimal choice is achieved by minimizing the K-fold cross-validation predictive criterion (see for instance Geisser [94], Breiman *et al.* [22], Burman [27], Clarke *et al.* [51], Hastie *et al.* [107], and Arlot and Celisse [5]). Our algorithm uses the following main steps :

for different values of h_s **do**
→ compute $\widehat{X}_1, \ldots, \widehat{X}_n$
→ split randomly into K roughly equal-sized blocks the whole sample ; for $k = 1, \ldots, K$, let $\{(X_i, Y_i);\, i \in I_k\}$ be the kth block where $I_1, \ldots, I_K$ is a partition of the set of subscripts $I := \{1, \ldots, n\}$

for different values of h_r **do**
for $k = 1$ to K **do**
→ compute $\widehat{\widehat{r}}^{-k}(\widehat{X}_i)$ for any $i \in I_k$, where $\widehat{\widehat{r}}^{-k}$ is the nested-kernel estimator obtained when removing the kth block from the whole sample (i.e. $\{(X_l, Y_l);\, l \in I \backslash I_k\}$)
→ derive the kth-block-based relative cross validation

$$CV_k(h_s,\, h_r) \;:=\; \Big[\sum_{i\in I_k}\Big\{Y_i - \widehat{\widehat{r}}^{-k}(\widehat{X}_i)\Big\}^2\Big] \,/\, \Big[\sum_{i\in I_k}\Big\{Y_i - \overline{Y}\Big\}^2\Big],$$

where $\overline{Y}$ stands for the average of the Y_i's over the kth block
end for
→ compute the global K-fold CV : $KCV(h_s, h_r) := K^{-1}\sum_{k=1}^{K} CV_k(h_s, h_r)$
end for
end for
→ compute the optimal bandwidths by minimizing the K-fold CV criterion :

$$(h_s^{opt},\, h_r^{opt}) \;=\; \arg\min_{h_s,h_r} KCV(h_s, h_r).$$

It is worth noting that according to the simple shape of the estimator, the whole estimating procedure is very easy to implement ; its practical behavior is studied in Section 4.5.

4.3 About supervised classification

As a by-product of the presented methodology, one can derive a similar estimating procedure in the supervised classification setting. Suppose now that our sample of curves $X_1, \ldots, X_n$ can be split into G known groups. This amounts to the situation where we observe a sample of n pairs $(X_1, L_1), \ldots, (X_n, L_n)$ iid $\sim (X, L)$, with, for $i = 1, \ldots, n$, $L_i \in \{1, 2, \ldots, G\}$ containing the class membership label of the ith curve X_i. Given the random curve X, we are interested in estimating the G class membership probabilities $p_1(X) = P(L = 1 \,|\, X)$, $p_2(X) = P(L = 2 \,|\, X), \ldots, p_G(X) = P(L = G \,|\, X)$ in order to predict the corresponding label as the one with the highest membership probability. For any classmembership g, estimating the conditional probability $P(L = g \,|\, X = x)$ from $(X_1, L_1), \ldots, (X_n, L_n)$ amounts to estimate the conditional expectation $p_g(x) = E\,\{Y \,|\, X = x\}$ from $(X_1, Y_1), \ldots, (X_n, Y_n)$ with $Y_i = 1_{L_i = g}$ for $i = 1, \ldots, n$. So, the supervised classification problem can be translated into g regression problems in order to estimate the g regression operators (i.e. conditional probabilities) $p_1, \ldots, p_G$. Consider now the errors-in-variables context where only a corrupted version $X^* = X + \eta$ of the random curve X is available. The nested-kernel estimator is still valid in this setting up to straightforward adaptation and hence can be used for computing, for $g = 1, \ldots, G$, the following probability :

$$\widehat{\widehat{p}}_g(x) \;=\; \frac{\sum_{\{i,\, L_i = g\}} K_r\left\{h_r^{-1}\|x - \widehat{X}_i\|_2\right\}}{\sum_{i=1}^{n} K_r\left\{h_r^{-1}\|x - \widehat{X}_i\|_2\right\}},$$

where the $\widehat{X}_i$'s are the smoothed version of the X_i's defined with (4.1). So, for given h_s, h_r and for any noisy curve X_i^*, one can estimate its class membership label : $\widehat{L}_i := \arg\max_{g\in\{1,\ldots,G\}} \widehat{\widehat{p}}_g(\widehat{X}_i)$. The previous algorithm can be adapted to this supervised classification problem by just replacing :
→ $\widehat{\widehat{r}}^{-k}$ with $\widehat{\widehat{p}}_1^{-k}, \ldots, \widehat{\widehat{p}}_G^{-k}$,
→ $CV_k(h_s, h_r)$ with the misclassification rate $MCR_k(h_s, h_r) := |I_k|^{-1}\sum_{i\in I_k} 1_{L_i \neq \widehat{L}_i}$,

$\rightarrow KCV(h_s, h_r)$ with $KMCR(h_s, h_r) := K^{-1} \sum_{k \in K} MCR_k(h_s, h_r)$.
The implementation of this algorithm can be found in Section 4.5.2 where an example of hyperspectral dataset is processed for classifying woody species.

4.4 Some asymptotic properties

The main challenge of this section is to state the theoretical behaviour of the nested-kernel estimator $\widehat{\widehat{r}}$ of the regression operator r in the nonparametric model $Y_i = r(X_i) + \varepsilon_i$ when the X_i's are observable up to a measurement error process η. So, one has at hand only a noisy version $X_i^*(\lambda) = X_i(\lambda) + \eta_i(\lambda)$ of X_i and the X_i^*'s are not observed continuously over the range Λ but only sampled at a grid $\lambda_1, \ldots, \lambda_d$. Now, we are ready to enumerate assumptions before formulating our main results.

4.4.1 Assumptions

From now on and without loss of generality, one sets $\Lambda = [0, 1]$. In addition, let C (possibly with subscript or upperscript) stands for any nonnegative constant.

— *Assumptions on the model.*

(H1) X is assumed to be a regular random process in that sense, for any $(\lambda, \lambda') \in [0, 1]^2$, the covariance function $Cov\{X(\lambda), X(\lambda')\} := c(\lambda, \lambda')$ is a twice differentiable function with respect to λ and λ',

(H2) r is a Lipschitz operator of order α : for all $(x_1, x_2) \in L^2_{[0,1]} \times L^2_{[0,1]}$, it exists C and $\alpha > 0$ such that $|r(x_1) - r(x_2)| \leq C \|x_1 - x_2\|_2^\alpha$,

(H3) the zero-mean process η of the measurement errors (i.e. $\forall \lambda \in [0, 1]$, $E\,\eta(\lambda) = 0$) has the following property : it exists a twice differentiable variance function $\sigma_\eta^2 \in L^2_{[0,1]}$ such that, for any $(\lambda, \lambda') \in [0, 1]^2$, $E\,\eta(\lambda)\,\eta(\lambda') = \sigma_\eta^2(\lambda)\,1_{\lambda=\lambda'}$,

(H4) the grid $0 = \lambda_0 < \lambda_1 < \cdots < \lambda_d = 1$ is equispaced and $d = d_n$ tends to ∞ with n.

— *Assumptions on the estimator.*

(H5) Let $\varphi_x(h_r) := P(\|x - X\|_2 < h_r)$ be the small ball probability associated to the ball of center x and radius h_r ; it exists $\gamma \in (1/3, 1)$, C and C' such that, for n large enough :

(i) $C\,\varphi_x(h_r) \leq \varphi_x(h_r + o(h_r)) \leq C'\,\varphi_x(h_r)$,

(ii) $a_n^{2\gamma} = o(\varphi_x(h_r))$ and $a_n^{1-\gamma} = o(h_r)$, where $a_n = h_s + 1/\sqrt{d\,h_s}$ is the univariate rate of convergence for the presmoothing step arising from Lemma 1,

(H6) h_s and h_r are two sequences depending on n such that, when n tends to infinity, h_s and h_r tend to zero whereas $d\,h_s$ and $n\,\varphi_x(h_r)$ tend to infinity,

(H7) K_s is a twice differentiable symmetric kernel with $supp(K_s) = (-1, 1)$; K_r is an asymmetrical kernel such that, for all v, it exists C and C', $C\,1_{[0,1]}(v) \leq K_r(v) \leq C'\,1_{[0,1]}(v)$.

Comments on assumptions. Although most of these assumptions are quite standard in the nonparametric setting, two of them, (H1) and (H5), deserve more attention. The first one, (H1), requires that the unobservable random function X has a covariance function regular enough. However, it seems to be reasonable since we just postulate that the underlying non noised process X is quite smooth, which is an essential hypothesis for deriving the rate of convergence in the presmoothing step (see Lemma 1). The second assumption, (H5), contains more technical hypotheses.

(H5-i) assumes that small ball probabilities of X are not sensitive to acceptable disruptions on the radius (see Li and Shao [142] for a survey on small ball probability). However, the class of random functions fulfilling (H5-i) is rather large. For instance, for any exponential-type process such that, for ϵ small enough, $\varphi_x(\epsilon) \sim C\,\epsilon^{p_1}\exp\{-C'/\epsilon^{p_2}\}$ or $\varphi_x(\epsilon) \sim C\,\exp\{C'\epsilon^{-p_1}(\log\epsilon)^{p_2}\}$ with $p_1 > 0$ and $p_2 > 0$, (H5-i) holds. Concerning (H5-ii), it gives some conditions connecting the sample size n with the grid size d and both bandwidths h_s and h_r. In the particular case where $h_s \sim d^{-1/3}$ (i.e. $h_s \sim 1/\sqrt{d\,h_s}$) and $d \sim n^{\frac{3}{2\gamma}+\delta}$ with $\delta > 0$, it is easy to see that :

— $a_n^{2\gamma} = o\big(\varphi_x(h_r)\big)$, keeping in mind that $n^{-1} = o\big(\varphi_x(h_r)\big)$ thanks to (H6),

— the condition $a_n^{1-\gamma} = o\big(h_r\big)$ amounts to assume that it exists $\rho \in (0, 1+2\delta/9)$, $n^{-\rho} = o\big(h_r\big)$, which is not a restrictive constraint.

This particular situation involves a number of measurements $d \sim n^b$ (sampled grid of the noisy curves) with $b > 3/2 + \delta$. In other words, d is much larger than the sample size n. It is worth noting that this high-dimensional setting is especially in accordance with our hyperspectral datasets (see Section 4.5.2).

4.4.2 Main theoretical results

The first Lemma gives the rate of convergence of the mean square error between the smoothed functional covariate and the non observable one.

Lemma 1 *Under assumptions (H1), (H3), (H4), and (H7) one gets :*

$$E\|\widehat{X}_1 - X_1\|_2^2 \;=\; \frac{C}{d\,h_s}\left\{\int_\Lambda \sigma_\eta^2(\lambda)\,d\lambda\right\} \;+\; O(h_s^2) \;+\; O\left(\frac{1}{d\,h_s}\right). \tag{4.2}$$

As a by-product of Lemma 1, it holds that $\|\widehat{X}_1 - X_1\|_2 = O_P(a_n)$ with $a_n = h_s + 1/\sqrt{d\,h_s}$. The presentation of this lemma is not usual because when focusing on the right side of (4.2), the last term encompasses the first one. However, this non optimal writing emphasizes the role played by the variance function σ_η^2 of the measurement errors process. It is clear that the mean square error $E\|\widehat{X}_1 - X_1\|_2^2$ increases with the quantity $\int_\Lambda \sigma_\eta^2(\lambda)\,d\lambda$.

The second lemma focuses on small ball probabilities based on the non observable functional covariate, named $P\left(\|x - X_i\|_2 < h_r\right)$, and its smoothed counterpart $P\left(\|x - \widehat{X}_i\|_2 < h_r\right)$. The terminology small ball refers to the situation when the parameter h_r is close to zero, which is the case in the considered framework (h_r tends to 0 with n). The following result gives conditions leading to comparable asymptotic behaviour for both these quantities.

Lemma 2 *Under (H1)-(H4) and (H6)-(H5), it exists C and C' such that*

$$C\,\varphi_x(h_r) \;\le\; P\left(\|x - \widehat{X}_i\|_2 \;<\; h_r\right) \;\le\; C'\,\varphi_x(h_r).$$

Lemma 2 is a crucial step in order to get the consistency of the nested-kernel estimator $\widehat{\widehat{r}}$ of the regression operator r which is the aim of next result.

Theorem 1 *If (H1)-(H7) are satisfied, one gets*

$$\widehat{\widehat{r}}(x) - r(x) \;=\; O(h_r^\alpha) + O_P\left(1/\sqrt{n\,\varphi_x(h_r)}\right).$$

This rate of convergence is the standard one obtained when observing non corrupted functional predictors. So, from an asymptotic viewpoint, $\widehat{\widehat{r}}$ is not negatively impacted by the denoising stage as soon as the corresponding rate of convergence a_n is high enough. The similar rate of convergence holds in the supervised classification as a straightforward consequence of Theorem 1. To this end, just replace (H2) with (H2') by substituting successively $p_1, \ldots, p_G$ for r.

Corollary 1 *Under (H1), (H2'), (H3)-(H7), it comes for* $g = 1, \ldots, G$:

$$\widehat{\widehat{p}}_g(x) - p_g(x) \;=\; O(h_r^{\alpha}) + O_P\left(1/\sqrt{n\,\varphi_x(h_r)}\right).$$

Extension. These theoretical developments can be generalized to the situation when the standard norm $\|.\|_2$ is replaced with any proximity measure $\Pi(\cdot,\cdot)$ in the nested-kernel estimator. All these results remains still valid as soon as, for all (u, v) in $L^2_\Lambda \times L^2_\Lambda$, it exists a nonnegative constant M such that $\Pi(u, v) \;\leq\; M\,\|u - v\|_2$. The motivation of this technical extension come from practical situations when other proximity indices between curves (for instance distance based on successive derivatives or derived from partial least squares analysis) are more relevant than the classical L_2-norm, which is often the case when considering hyperspectral datasets.

4.5 Nested-kernel estimator in action

This section is devoted to the implementation of the nested-kernel estimator in order to highlight its finite sample properties. All routines are programmed with the R language [170] and are available on request. They will be soon available online.

4.5.1 Simulated datasets

The main goal of this section is to assess finite-sample properties of the nested-kernel. To this end, we use the 5-fold cross-validation criterion 5-CV introduced in Section 4.2 and the following scheme for simulating datasets.

Simulating underlying smooth functional predictors. Let $X_1, \ldots, X_{500}$ be i.i.d. functional predictors (without contamination) sampled at d equispaced measurements $\lambda_1, \ldots, \lambda_d$ in the range $\Lambda = [-1,\, 1]$ such that, for $i = 1, \ldots, 500$, $X_i(\lambda_j) \;=\; A_{1,i}\,\cos(2\pi\lambda_j/3) \;+\; A_{2,i}\,\sin(5\pi\lambda_j/7) \;+\; 0.5\,A_{3,i}\,\left\{1 - \exp(\lambda_j^2)\right\}$ where the $A_{1,i}$'s (resp. $A_{2,i}$'s and $A_{3,i}$'s) are 500 iid r.r.v. generated from a uniform distribution on $[-1,\, 1]$.

Building scalar responses. From the X_i's, corresponding scalar responses are built according to three regression models : $Y_i \;=\; r_k(X_i) \;+\; \varepsilon_i$ where $r_1(X_i) \;=\; \int_{-1}^{1} \exp\left\{-X_i^2(\lambda)\right\} d\lambda$ (model 1), $r_2(X_i) \;=\; \int_{-1}^{1} \sin\left\{1.5\,X_i^2(\lambda)\right\} d\lambda$ (model 2), and $r_3(X_i) \;=\; \int_{-1}^{1} \exp\left\{-X_i^2(\lambda)\right\} \sin\left\{1.5\,X_i^2(\lambda)\right\} d\lambda$ (model 3), with the ε_i's iid as a centered Gaussian with variance equals to $0.05 \times var\{r_k(X)\}$.

Contamining functional predictors. For $i = 1, \ldots, 500$ and $j = 1, \ldots, d$, let $X^*_{i,k}(\lambda_j) = X_i(\lambda_j) + \eta_i^k(\lambda_j)$ be a contaminated version of $X_i(\lambda_j)$ with, for $k = 1, \ldots, 4$, the measurement errors $\eta_i^k(\lambda_j)$ independently generated from $N(0, \sigma^2_{i,k}(\lambda_j))$ such that $\sigma^2_{i,1}(\lambda_j) \;=\; c_1 \times var\{X(\lambda_j)\}$ (i.e. constant noise-to-signal ratio w.r.t. i and $j \rightarrow$ scenario 1), $\sigma^2_{i,2}(\lambda_j) \;=\; c_2 \exp(-5\lambda_j^2)$ (i.e. varying noise-to-signal ratio with j : the highest variance occurs at 0 and decreases exponentially up to the bounds $\rightarrow$ scenario 2), $\sigma^2_{i,3}(\lambda_j) \;=\; c_3\{1_{[-1,\,0]}(\lambda_j) + 10 \times 1_{(0,\,1]}(\lambda_j)\}$ (i.e. dichotomous noise-to-signal ratio w.r.t j : the variance is 10 times higher for the second part than for the first one $\rightarrow$ scenario 3), $\sigma^2_{i,4}(\lambda_j) \;=\; c_4\{1_{[-1,\,U_i]\cup(U_i+1,\,1]}(\lambda_j) + 10 \times 1_{(U_i,\,U_i+1]}(\lambda_j)\}$ (i.e. dichotomous noise-to-signal ratio w.r.t. j and varying with i : the highest variance occurs on the half part starting randomly at U_i, where the U_i's are iid r.r.v. uniformly distributed on $[-1, 0) \rightarrow$ scenario 4). For $k = 1, \ldots, 4$, the positive constant c_k is set in order to control the averaged noise-to-signal ratio $\overline{nsr} := d^{-1}\sum_j var\{\eta^k(\lambda_j)\}/var\{X(\lambda_j)\}$ over the whole set of measurements. Figure 4.1 gives an idea on the various proposed contamination scenarios with $d = 100$ and $\overline{nsr} = 2$. For scenarios 3 and 4, vertical lines mark off the areas with highest noise-to-signal.

Denoising with a basic local bandwidth. Scenarios 2-4 involve structures of noise which vary with the measurements. So, in order to build a bandwith depending on the $\lambda'_j s$, instead of using a

			$\overline{nsr} = 0.5$	$\overline{nsr} = 1$	$\overline{nsr} = 1.5$	$\overline{nsr} = 2$
scenario 1	model 1	NKE	0.136 (0.010)	0.174 (0.015)	0.211 (0.017)	0.239 (0.019)
		KE	0.210 (0.008)	0.277 (0.020)	0.393 (0.072)	0.530 (0.034)
	model 2	NKE	0.136 (0.010)	0.179 (0.017)	0.216 (0.017)	0.252 (0.019)
		KE	0.212 (0.015)	0.271 (0.020)	0.396 (0.027)	0.54 (0.036)
	model 3	NKE	0.196 (0.019)	0.260 (0.025)	0.316 (0.032)	0.358 (0.032)
		KE	0.318 (0.027)	0.408 (0.025)	0.564 (0.086)	0.693 (0.032)
scenario 2	model 1	NKE	0.130 (0.009)	0.159 (0.013)	0.183 (0.012)	0.210 (0.013)
		KE	0.230 (0.021)	0.318 (0.020)	0.442 (0.030)	0.597 (0.052)
	model 2	NKE	0.131 (0.010)	0.167 (0.017)	0.193 (0.016)	0.213 (0.019)
		KE	0.226 (0.016)	0.325 (0.022)	0.468 (0.059)	0.606 (0.055)
	model 3	NKE	0.184 (0.020)	0.239 (0.022)	0.280 (0.028)	0.315 (0.037)
		KE	0.349 (0.022)	0.467 (0.029)	0.621 (0.023)	0.750 (0.066)
scenario 3	model 1	NKE	0.130 (0.009)	0.165 (0.013)	0.191 (0.014)	0.220 (0.020)
		KE	0.218 (0.017)	0.319 (0.057)	0.430 (0.030)	0.574 (0.024)
	model 2	NKE	0.130 (0.010)	0.170 (0.012)	0.194 (0.017)	0.222 (0.021)
		KE	0.215 (0.014)	0.309 (0.021)	0.429 (0.029)	0.581 (0.033)
	model 3	NKE	0.193 (0.019)	0.236 (0.023)	0.282 (0.028)	0.319 (0.027)
		KE	0.333 (0.022)	0.469 (0.095)	0.581 (0.033)	0.735 (0.036)
scenario 4	model 1	NKE	0.130 (0.009)	0.170 (0.011)	0.204 (0.015)	0.236 (0.020)
		KE	0.218 (0.015)	0.295 (0.020)	0.423 (0.031)	0.575 (0.033)
	model 2	NKE	0.134 (0.009)	0.175 (0.017)	0.208 (0.018)	0.237 (0.020)
		KE	0.226 (0.022)	0.306 (0.020)	0.430 (0.027)	0.597 (0.029)
	model 3	NKE	0.194 (0.020)	0.248 (0.029)	0.303 (0.034)	0.350 (0.035)
		KE	0.338 (0.029)	0.455 (0.068)	0.585 (0.029)	0.743 (0.031)

TABLE 4.1 – 5-CV values with corresponding standard deviations in brackets.

constant h_s, one proposes to define $h_{s,j} \propto \widehat{\sigma}_j$ where $\widehat{\sigma}_j$ is the standard deviation of the contaminated functional predictor valued at λ_j named $X^*(\lambda_j)$.

For each combination of regression model, contamination scenario and $\overline{nsr}$ level, the previous simulation scheme is repeated 50 times leading to 50 values of 5-CV. Table 4.1 sums up the results obtained for $d = 100$, where NKE (i.e. 5-CV_{NKE}) stands for our nested-kernel estimator with standard L_2-norm as proximity measure between curves and KE denotes the standard functional kernel estimator without the denoising stage. Overall, the predictive performances of NKE depends linearly on the level of $\overline{nsr}$, which is not the case for KE and the relative gain (5-CV_{KE} − 5-CV_{NKE})/5-CV_{KE} increases with $\overline{nsr}$. The relative gain varies from 33% to 63% where the highest gain is reached for the highest averaged noise-to-signal ratio. For model 1 and 2 of comparable complexity, the predictive results are very similar whereas for model 3 (which is of higher complexity) one can observe a slight degraded performance. About contamination scenarios, it appears that there is no trivial impact on the predictive criterion, although the relative gain seems to be slightly larger for non uniform contamination (i.e. scenarios 2, 3 and 4). Now, to better assess the NKE's predictive performance, regression models are estimated by using the simulated underlying smooth functional predictors (i.e. the unobservable curves $X_1, X_2, \ldots$). The following 5-CV values with standard deviation in brackets are obtained : 0.094 (0.007) for model 1, 0.088 (0.006) for model 2 and 0.012 (0.011) for the model 3.

Although NKE produces predictive performances not so far from the best ones for $\overline{nsr} = 0.5$,

		scenario 1	scenario 2	scenario 3	scenario 4
model 1	NKE	0.116 (0.009)	0.118 (0.009)	0.120 (0.009)	0.115 (0.007)
	KE	0.377 (0.047)	0.455 (0.044)	0.373 (0.047)	0.428 (0.041)
model 2	NKE	0.115 (0.008)	0.114 (0.008)	0.116 (0.010)	0.116 (0.009)
	KE	0.385 (0.046)	0.485 (0.046)	0.415 (0.066)	0.436 (0.047)
model 3	NKE	0.167 (0.013)	0.168 (0.015)	0.170 (0.015)	0.169 (0.013)
	KE	0.536 (0.039)	0.633 (0.095)	0.562 (0.048)	0.623 (0.041)

TABLE 4.2 – 5-CV values with corresponding standard deviations in brackets when $d = 1000$ and $\overline{nsr} = 2$.

it is definitely not the case for higher noise-to-signal ratios. According to the theory, the rate of convergence of the nested-kernel estimator to the true one should not be impacted by the denoising stage as soon as the univariate rate of convergence a_n (see H5-ii and related comments) is high enough. However, we are here in the setting when the number of measurements $d = 100$ is smaller than the sample size $n = 500$ and the univariate rate of convergence of the denoising stage may be relatively low in this situation. Consequently, it may degrade the overall rate of convergence of the nested-kernel estimator. In the opposite, one can expect to increase predictive performances with a better rate a_n. To illustrate this purpose, we use an analogous simulation scheme but with $d = 1000$ (instead of $d = 100$) because a larger number of measurements should improve the univariate rate of convergence of the denoising stage (i.e. a_n). Table 4.2 gives the predictive performances for the worst situation corresponding to $\overline{nsr} = 2$. Clearly, increasing d (and hence improving the univariate rate of convergence of the denoising stage a_n) allows to improve significantly the predictive performance of the nested-kernel estimator ; the new results for $\overline{nsr} = 2$ are even better than those obtained previously with $d = 100$ and $\overline{nsr} = 0.5$. In addition, the relative gain varies now between 68% and 76% (instead of 33% and 63% for $d = 100$).

4.5.2 Application to hyperspectral dataset

4.5.2.1 Regression setting : a pseudo hyperspectral dataset based on physical modelling

This section focuses on a pseudo hyperspectral dataset coming from the teledetection community. The idea of its conceptor was to build hyperspectra by using biophysical variables (i.e. responses) where the relationship between hyperspectra and reflectance is based on biophysical model (see Jacquemoud *et al.* [116]) and not on some statistical modelling. Consequently, the statistical link between responses and hyperspectra is not known. The biophysical variable of interest in our study is the *chlorophyll* content (given in percentage). The overall dataset contains 5000 hyperspectra sampled at 2101 wavelengths ranging from 400 to 2500 nm (spectral resolution equals to 1 nm). Figure 4.2 (a) displays a sample of 5 pseudo hyperspectra (without contamination).

The main aim of this study is to assess the ability of our nested-kernel estimator (NKE) in predicting the chlorophyll content from a contaminated version of these pseudo hyperspectra according to different settings. Two sampling frequency situations for the hyperspectra are considered : the high frequency corresponds to the original hyperspectra (i.e. $d = 2101$ equispaced wavelengths) whereas the low frequency sampling involves only $d = 211$ equispaced wavelengths (one wavelength over ten is retained). In order to be as close to reality as possible, we consider

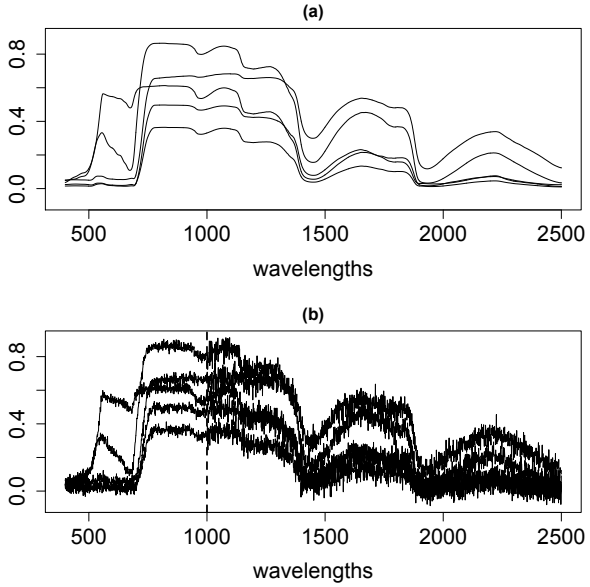


FIGURE 4.2 – (a) Sample of 5 pseudo hyperspectra built by using biophysical properties and (b) corresponding contaminated ones when using dichotomous noise with $\overline{nsr} = 0.2$.

the dichotomous contamination scenario 3 used in Section 4.5.1. Indeed, the sensors allowing to build the whole hyperspectra are generally divided into two parts, one part collecting informations from 400 nm to 1000 nm and the second registering data for the remaining wavelengths. Based on sensor properties, the noise-to-signal ratio is systematically higher for the second spectrum part. Figure 4.2 (b) displays corresponding noisy hyperspectra. The vertical line divides the wavelengths into two ranges ; the left (resp. right) one corresponds to a low (resp. high) noise-to-signal ratio. The NKE procedure is launched in order to predict chlorophyll (Y) contents from contaminated pseudo hyperspectra. 500 hyperspectra and corresponding scalar responses $\{X_i, Y_i\}_{i=1,\ldots,500}$ are randomly selected to build NKE $\widehat{\widehat{r}}$ and the 5-CV value (i.e. in-sample). An additional dataset $\{X_i, Y_i\}_{i=501,\ldots,1000}$ (i.e. out-sample) is used for computing the relative mean square error $RMSEP = \sum_{i=501}^{1000}\{Y_i - \widehat{\widehat{r}}(X_i)\}^2 / \sum_{i=501}^{1000}(Y_i - \overline{Y})^2$. This simulation scheme is repeated 50 times. At this stage, it is worth noting that a distance based on the first derivative of the hyperspectra in the NKE (instead of the classical L_2-norm between original curves) was implemented in order to improve the predictive quality. More details on such useful proximity indices can be found in [87] (Chapter 3). Table 4.3 gathers predictive performances in terms of 5-CV as well as $RMSEP$ according to two sampling frequencies and three averaged noise-to-signal levels ($\overline{nsr}$) : 5%, 10%, and 20%.

The same conclusions than those obtained in Section 4.5.1 hold : predictive performances increase with the frequency of the sampling (i.e. the number of wavelengths d) and the gain of NKE w.r.t. KE goes up with the averaged noise-to-signal level.

4.5.2.2 Application to supervised classification hyperspectral problem

In order to assess the performance of our nested-kernel estimator in a supervised classification hyperspectral problem, one considers a real hyperspectral image, called *MADONNA*, collected on the site of Villelongue, France, by the HYSPEX sensors. The data consist in 32224 pixels (with a spatial resolution of 50 cm), each pixel corresponding to the observation of one hyperspectrum with 160 spectral bands from 400 to 1000 nm (i.e. the number of measurements d equals to

		$d = 211$	$d = 2101$
$\overline{nsr} = 0.05$	NKE	0.147 (0.011) / **0.149 (0.013)**	0.123 (0.010) / **0.128 (0.014)**
	KE	0.247 (0.021) / **0.236 (0.021)**	0.170 (0.012) / **0.163 (0.014)**
$\overline{nsr} = 0.1$	NKE	0.171 (0.015) / **0.169 (0.012)**	0.135 (0.013) / **0.143 (0.012)**
	KE	0.331 (0.025) / **0.313 (0.021)**	0.214 (0.020) / **0.209 (0.014)**
$\overline{nsr} = 0.2$	NKE	0.208 (0.019) / **0.209 (0.015)**	0.148 (0.012) / **0.153 (0.015)**
	KE	0.463 (0.033) / **0.439 (0.035)**	0.278 (0.018) / **0.264 (0.019)**

TABLE 4.3 – 5-CV and ***RMSEP*** values with corresponding standard deviations in brackets ; $d = 211$ (resp. $d = 2101$) $\leftrightarrow$ low (resp. high) sampling frequency.

NKE	KE	GMM	SVM	Random Forest
0.105	0.131	0.149	0.138	0.185
(0.010)	(0.011)	(0.013)	(0.014)	(0.012)

TABLE 4.4 – Averaged misclassification rates over 50 repetitions with corresponding standard deviations in parentheses.

160). The following 12 woody species have been identified during a field campaign : Ash (4333), Beech (42), Birch (468), Chestnut (2855), Fern (1983), Goat willow (485), Hazel (4122), Linden (3402), Locust (2372), Maple (165), Oak (10981) and Walnut (1016). The number in parentheses following each category refers to the number of corresponding observed hyperspectra. Figure 4.3 gives an idea on the hyperspectra profiles for each woody species. In reality, a field campaign is so expensive so that it is impossible to identify the whole set of pixels. So, the aim of this study is to assess the ability of NKE in predicting the 12 woody species from a learning sample of hyperspectra with a reasonable size.

This is why in our experimental protocol we consider only 30 pixels (i.e. 30 hyperspectra) per category. Thus, for each woody species, the class membership probability NKE is computed in the situation where $n = 30$ and $d = 160$. In order to assess the ability of NKE to predict the woody species, the overall misclassification rate for the 31864 remaining pixels is worked out (i.e. the 31864 remaining hyperspectra are assigned to the class membership of highest predicted probability). Note that the best predictive performances are obtained when NKE uses a distance based on partial least squares (PLS) decomposition (for more details on PLS, see for instance [214], [112], [153] ; for implementing PLS-based distance in the functional nonparametric regression framework, see [87], Chapter 3). 50 learning samples of 360 hyperspectra with corresponding class membership are randomly generated in order to get 50 misclassification rates for the remaining pixels. Alternative methods have been implemented in order to assess the predictive performances of NKE : the functional nonparametric discrimination KE (i.e. NKE without the embedded presmoothing stage), and three multivariate methods named the Gaussian Mixture Model (GMM, [106]), the Support Vector Machine (SVM, [209]), and the Random Forest method ([21]). In order to get a fair comparison, GMM, SVM and Random Forest are adapted to this functional setting by expanding the hyperspectra into a B-spline basis. The hyperspectra are replaced by the coefficients coming from the basis expansion and the three multivariate methods are applied on the coefficient matrix. The size of the B-spline basis is selected in order to minimize the in-sample misclassification rate. The results are reported in Table 4.4.

When comparing NKE with KE, the presmoothing stage contributes significantly to the

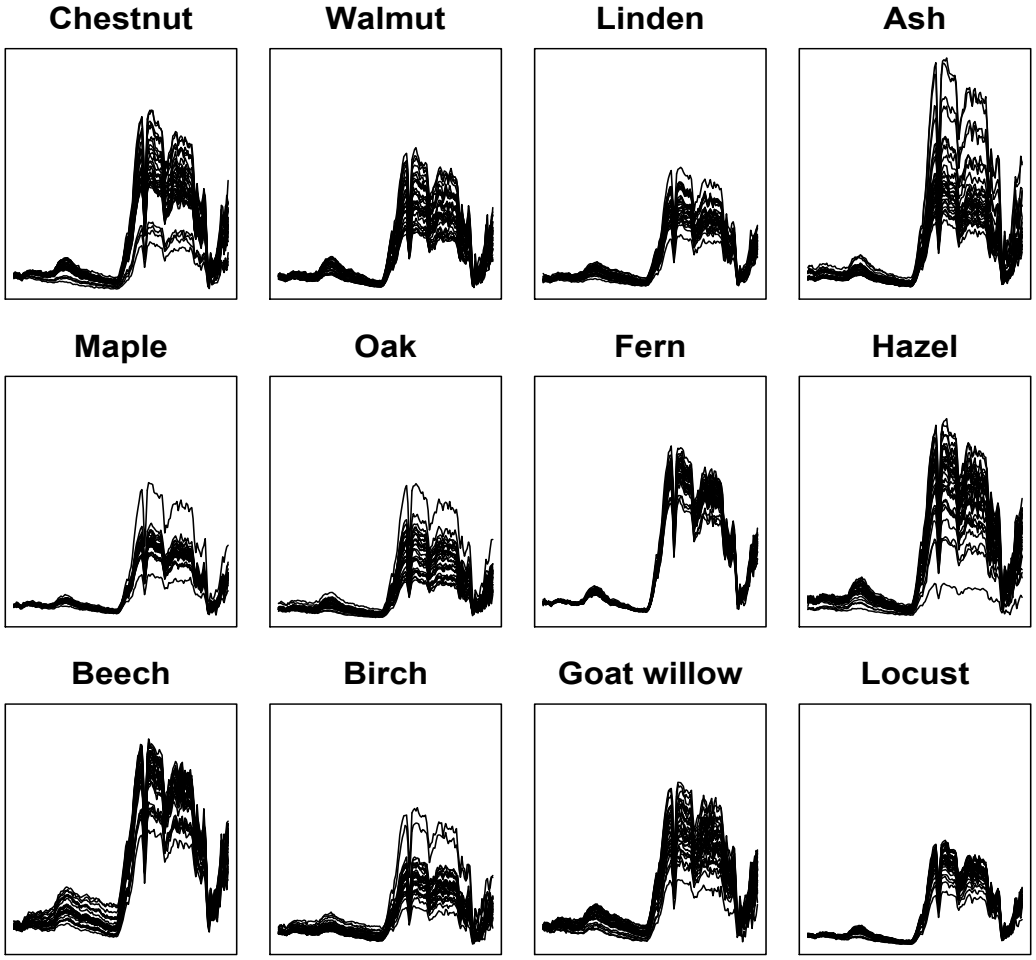


FIGURE 4.3 – Sample of 30 hyperspectra for each woody species of *MADONNA* dataset (the same scale is involved in each plot).

misclassification rate improvement. One observes also a gain greater than 25 percent with respect to the other methods (which corresponds to a correct assignment of around 1200 supplementary pixels).

4.6 Acknowledgments

This research was supported in part by the French National Spacial Agency (CNES) and the Midi-Pyrénées region.

4.7 Appendix : proofs of lemmas and theorem

We remind that C (possibly with subscript or upperscript) stands for any nonnegative generic constant.

Proof of Lemma 1. For $j = 0, \ldots, d$, set $w_j(\lambda) = K_s\{(\lambda - \lambda_j)/h_s\} \;/\; \sum_{j=0}^{d} K_s\{(\lambda - \lambda_j)/h_s\}$.

As $\widehat{X}_i(\lambda) = \sum_{j=0}^d w_j(\lambda)\, X^*(\lambda_j)$ with $\sum_{j=0}^d w_j(\lambda) = 1$, one can write

$$\begin{aligned}\left\{\widehat{X}_i(\lambda) - X_i(\lambda)\right\}^2 &= \left\{\sum_{j=0}^d \left[X_i^*(\lambda_j) - X_i(\lambda)\right] w_j(\lambda)\right\}^2 \\ &= \sum_{j=0}^d \sum_{j'=0}^d \left\{X_i^*(\lambda_j) - X_i(\lambda)\right\} \left\{X_i^*(\lambda_{j'}) - X_i(\lambda)\right\} w_j(\lambda) w_{j'}(\lambda).\end{aligned}$$

Using the last equality, it comes

$$\|\widehat{X}_i - X_i\|_2^2 = \int_0^1 \sum_{j=0}^d \sum_{j'=0}^d \left\{X_i^*(\lambda_j) - X_i(\lambda)\right\} \left\{X_i^*(\lambda_{j'}) - X_i(\lambda)\right\} w_j(\lambda)\, w_{j'}(\lambda)\, d\lambda.$$

Now, by replacing X_i^* with $X_i + \eta$, one gets :

$$E\,\|\widehat{X}_i - X_i\|_2^2 = \int_0^1 \sum_{j=0}^d \sum_{j'=0}^d E\,(T_1 + T_2 + T_3 + T_4)\, w_j(\lambda)\, w_{j'}(\lambda)\, d\lambda, \tag{4.3}$$

where $T_1 = \{X_i(\lambda_j) - X_i(\lambda)\}\{X_i(\lambda_{j'}) - X_i(\lambda)\}$, $T_2 = \eta_i(\lambda_j)\{X_i(\lambda_{j'}) - X_i(\lambda)\}$, $T_3 = \eta_i(\lambda_{j'})\{X_i(\lambda_j) - X_i(\lambda)\}$ and $T_4 = \eta_i(\lambda_j)\,\eta_i(\lambda_{j'})$.

— Let us first remark that

$$E\,T_2 \;=\; E\,T_3 \;=\; 0. \tag{4.4}$$

— $E\,T_1 = c(\lambda_j, \lambda_{j'}) - c(\lambda_j, \lambda) - c(\lambda, \lambda_{j'}) + c(\lambda, \lambda)$. According to (H1),

$$c(\lambda + h_s\, v, \lambda') = c(\lambda, \lambda') + h_s\, v\, \frac{\partial c(\lambda, \lambda')}{\partial \lambda} + \frac{1}{2}\, h_s^2\, v^2\; \left.\frac{\partial^2 c(\lambda, \lambda')}{\partial \lambda^2}\right|_{\lambda=\theta} \tag{4.5}$$

for given $(\lambda, \lambda') \in (0,1)^2$ and some θ between λ and $\lambda + h_s\, v$; the same holds when considering the second argument λ'. By definition of the $w_j(\lambda)$'s, and using standard numerical integration results (see for instance Davis and Rabinowitz [56] or Evans [67]), one has

$$\begin{aligned}\sum_{j=0}^d c(\lambda_j, \lambda_{j'})\, w_j(\lambda) &= \frac{d^{-1} \sum_{j=0}^d c(\lambda_j, \lambda_{j'})\, K_s\left\{h_s^{-1}(\lambda - \lambda_j)\right\}}{d^{-1} \sum_{j=0}^d K_s\left\{h_s^{-1}(\lambda - \lambda_j)\right\}} \\ &= \frac{\int_0^1 c(u, \lambda_{j'})\, K_s\left\{h_s^{-1}(\lambda - u)\right\} du \;+\; O(d^{-1})}{\int_0^1 K_s\left\{h_s^{-1}(\lambda - u)\right\} du \;+\; O(d^{-1})} \\ &= \frac{h_s \int_{-\lambda/h_s}^{(1-\lambda)/h_s} c(\lambda + h_s\, v, \lambda_{j'})\, K_s(v) dv \;+\; O(d^{-1})}{h_s \int_{-\lambda/h_s}^{(1-\lambda)/h_s} K_s(v) dv \;+\; O(d^{-1})}.\end{aligned}$$

where the last equality is obtained after substitution. (H7) and (4.5) entail $\sum_{j=0}^d c(\lambda_j, \lambda_{j'})\, w_j(\lambda) \;=$

$c(\lambda,\lambda_{j'})+O(h_s^2)+O(d^{-1}h_s^{-1})$ for n large enough. By repeating successively similar arguments, it comes

$$\sum_{j=0}^{d}\sum_{j'=0}^{d} c(\lambda_j,\lambda_{j'})\,w_j(\lambda)\,w_{j'}(\lambda) \;=\; c(\lambda,\lambda)+O(h_s^2)+O(d^{-1}h_s^{-1}),$$

as well as $\sum_{j=0}^{d}\sum_{j'=0}^{d} c(\lambda_j,\lambda)\,w_j(\lambda)\,w_{j'}(\lambda) \;=\; c(\lambda,\lambda)+O(h_s^2)+O(d^{-1}h_s^{-1})$ and $\sum_{j=0}^{d}\sum_{j'=0}^{d} c(\lambda,\lambda_{j'})\,w_j(\lambda)\,w_{j'}(\lambda) = c(\lambda,\lambda)+O(h_s^2)+O(d^{-1}h_s^{-1})$, which implies :

$$\int_0^1 \sum_{j=0}^{d}\sum_{j'=0}^{d} E\,(T_1)\,w_j(\lambda)\,w_{j'}(\lambda)\,d\lambda \;=\; O(h_s^2)+O(d^{-1}h_s^{-1}). \tag{4.6}$$

— $E\,(T_4) \;=\; \sigma_\eta^2(\lambda_j)\,1_{j=j'}$ entails that

$$\int_0^1 \sum_{j=0}^{d}\sum_{j'=0}^{d} E\,(T_4)\,w_j(\lambda)\,w_{j'}(\lambda)\,d\lambda \;=\; \int_0^1 \sum_{j=0}^{d} \sigma_\eta^2(\lambda_j)\,w_j(\lambda)^2\,d\lambda. \tag{4.7}$$

As $\sum_{j=0}^{d}\sigma_\eta^2(\lambda_j)\,w_j(\lambda)^2 \;=\; d^{-1}\,\dfrac{d^{-1}\sum_{j=0}^{d}\sigma_\eta^2(\lambda_j)\,\left[K_s\left\{h_s^{-1}(\lambda-\lambda_j)\right\}\right]^2}{\left[d^{-1}\sum_{j=0}^{d}K_s\left\{h_s^{-1}(\lambda-\lambda_j)\right\}\right]^2}$, by using similar numerical integration and substitution arguments as previously, one is able to state, for n large enough :

$$\begin{aligned}\sum_{j=0}^{d}\sigma_\eta^2(\lambda_j)\,w_j(\lambda)^2 \;&=\; d^{-1}\,\frac{h_s\,\sigma_\eta^2(\lambda)\int_{-1}^{1}K_s(v)^2\,dv \;+\; O(h_s^3) \;+\; O(d^{-1})}{h_s^2 \;+\; O(h_s/d) \;+\; O(d^{-2})}\\ &=\; C\,\frac{\sigma_\eta^2(\lambda)}{d\,h_s} \;+\; O(d^{-2}h_s^{-2}) \;+\; O(h_s/d).\end{aligned} \tag{4.8}$$

Now, (4.3), (4.4) and (4.6)-(4.8) imply the claimed result

$$E\,\|\widehat{X}_i - X_i\|_2^2 = \frac{C}{d\,h_s}\left\{\int_0^1 \sigma_\eta^2(\lambda)\,d\lambda\right\} \;+\; O(h_s^2) \;+\; O(d^{-1}h_s^{-1}). \tag{4.9}$$

Proof of Lemma 2. According to Lemma 1, we remind that $E\,\|\widehat{X}_i - X_i\|_2^2 \;=\; O(a_n^2)$ with $a_n = h_s + 1/\sqrt{dh_s}$. Then, for any $C>0$ and $\gamma\in(1/3,1)$, one can write

$$P\left(\left\{\|\widehat{X}_i - X_i\|_2 > C\,a_n^{1-\gamma}\right\}\right) \;\leq\; M\,a_n^{2\gamma}. \tag{4.10}$$

Consider now the event $A \;=\; \left\{\|x-\widehat{X}_i\|_2 < h_r\right\}$ and set $A_0 \;=\; \left\{\|X_i-\widehat{X}_i\|_2 > Ca_n^{1-\gamma}\right\}$. It is clear that $A \;=\; A_1\cup A_2$ with $A_1 \;=\; A\cap\overline{A_0}$ and $A_2 \;=\; A\cap A_0$. By remarking that $A_1 \subset \left\{\|x-X_i\|_2 \leq h_r + Ca_n^{1-\gamma}\right\}$, one has $P\,(A_1) \;\leq\; \varphi_x\left(h_r + Ca_n^{1-\gamma}\right)$ whereas (4.10) implies that $P\,(A_2) \;\leq\; M\,a_n^{2\gamma}$. This leads to the following inequality :

$$P\left(\left\{\|x-\widehat{X}_i\|_2 < h_r\right\}\right) \;\leq\; \varphi_x\left(h_r + Ca_n^{1-\gamma}\right) \;+\; M\,a_n^{2\gamma}. \tag{4.11}$$

Let us focus now on the event $B = \left\{ \|x - \widehat{X}_i\|_2 < h_r - C a_n^{1-\gamma} \right\}$. As $B = B_1 \cup B_2$ with $B_1 = B \cap \overline{A_0}$ and $B_2 = B \cap A_0$, it is easy to see that $P(B) \leq P(A) + P(A_0)$, which entails with (4.10) and (4.11) :

$$\varphi_x\left(h_r - C a_n^{1-\gamma}\right) - M a_n^{2\gamma} \leq P(A) \leq \varphi_x\left(h_r + C a_n^{1-\gamma}\right) + M a_n^{2\gamma}. \tag{4.12}$$

Now (H5) allows to conclude : $C\,\varphi_x(h_r) \leq P\left(\left\{\|x - \widehat{X}_i\|_2 < h_r\right\}\right) \leq C'\,\varphi_x(h_r)$.

Proof of Theorem 1. Once Lemmas 1 and 2 established, the proof of the convergence of the nested estimator follow similar steps as in Ferraty and Vieu [87] but with additional refinements due to the presmoothing stage. Let us first note $\widehat{\widehat{r}} := \widehat{\widehat{r}}_N / \widehat{\widehat{r}}_D$ where $\widehat{\widehat{r}}_N := n^{-1} \sum_{i=1}^n Y_i \widehat{\Delta}_i$, $\widehat{\widehat{r}}_D := n^{-1} \sum_{i=1}^n \widehat{\Delta}_i$, $\widehat{\Delta}_i := \widehat{K}_i / E\,\widehat{K}_i$ with $\widehat{K}_i := K_r\left\{h_r^{-1} \|x - \widehat{X}_i\|_2\right\}$ for $i = 1, \ldots, n$. Based on the usual decomposition

$$\begin{aligned} \widehat{\widehat{r}}(x) - r(x) &= \frac{1}{\widehat{\widehat{r}}_D(x)} \Big\{ \underbrace{\widehat{\widehat{r}}_N(x) - E\,\widehat{\widehat{r}}_N(x)}_{Q_1} + \underbrace{E\,\widehat{\widehat{r}}_N(x) - r(x)}_{Q_2} \Big\} \\ &\quad - \frac{r(x)}{\widehat{\widehat{r}}_D(x)} \Big\{ \underbrace{\widehat{\widehat{r}}_D(x) - 1}_{Q_3} \Big\}, \end{aligned} \tag{4.13}$$

the stated rate of convergence of the nested-kernel estimator holds as soon as $Q_1 = O_P\left(1 / \sqrt{n\,\varphi_x(h_r)}\right)$, $Q_2 = O\left(h_r^\alpha\right)$ and $Q_3 = O_P\left(1 / \sqrt{n\,\varphi_x(h_r)}\right)$. Before going on, let us state the following technical lemma.

Lemma 3 *Under assumptions (H1), (H3)-(H4), and (H5-ii), it exists a nonnegative C such that, for n large enough, $P(\|x - X_1\| < C\,h_r) = 1$.*

According to the definition of Q_1, one has $E\,Q_1^2 \leq n^{-1}\,E\,Y_1^2\,\widehat{\Delta}_1^2$ with

$$\begin{aligned} E\,Y_1^2\,\widehat{\Delta}_1^2 &= E\,r(X_1)^2 \widehat{\Delta}_1^2 + E\,\varepsilon_1^2 \widehat{\Delta}_1^2 + 2\,E\,\varepsilon_1 r(X_1) \widehat{\Delta}_1^2 \\ &\leq E\left\{r(X_1)^2 - r(x)^2\right\} \widehat{\Delta}_1^2 + r(x)^2\,E\,\widehat{\Delta}_1^2 + C\,\varphi_x(h_r) \\ &\leq C\left\{E\,\|x - X_1\|_2^\alpha \widehat{\Delta}_1^2 \left\{r(X_1) - r(x) + 2r(x)\right\} + \varphi_x(h_r)^{-1}\right\} \\ &\leq C\left\{E\,\|x - X_1\|_2^{2\alpha} \widehat{\Delta}_1^2 + 2\,r(x)\,E\,\|x - X_1\|_2^\alpha \widehat{\Delta}_1^2 + \varphi_x(h_r)^{-1}\right\} \\ &\leq C\left\{h_r^{2\alpha}\,\varphi_x(h_r)^{-1} + h_r^\alpha\,\varphi_x(h_r)^{-1} + \varphi_x(h_r)^{-1}\right\}, \end{aligned}$$

the last inequality combining the results of Lemmas 2 and 3. Finally, it comes that $E\,Q_1^2 \leq C\,n^{-1}\,\varphi_x(h_r)^{-1}$, which entails :

$$Q_1 = O_P\left(1 / \sqrt{n\,\varphi_x(h_r)}\right). \tag{4.14}$$

Let us focus now on the bias term :

$$\begin{aligned} Q_2 &= E\,Y_1\,\widehat{\Delta}_1 - r(x) \\ &= (E\,\widehat{K}_1)^{-1}\,E\left\{(Y_1 - r(x))\,\widehat{K}_1\right\} \\ &= (E\,\widehat{K}_1)^{-1}\,E\left\{(r(X_1) - r(x))\,\widehat{K}_1\right\} + (E\,\widehat{K}_1)^{-1}\,E\,\varepsilon_1 \widehat{K}_1 \\ &\leq (E\,\widehat{K}_1)^{-1}\,E\,\|x - X_1\|_2^\alpha \widehat{K}_1. \end{aligned}$$

By using Lemma 3, it holds :

$$Q_2 \,=\, O\left(h_r^{\alpha}\right). \tag{4.15}$$

By remarking that $E\,\widehat{\Delta}_i \,=\, 1$, $E\,Q_3^2 \,=\, E\left\{n^{-1}\sum_{i=1}^{n}\left(\widehat{\Delta}_i - 1\right)\right\}^2 \,\leq\, n^{-1}\,E\,\widehat{\Delta}_1^2$. By involving Lemma 2 with (H7), it is easy to see that $C\,\varphi_x(h_r) \,\leq\, E\,\widehat{K_1} \,\leq\, C'\,\varphi_x(h_r)$ and $C\,\varphi_x(h_r) \,\leq\, E\,\widehat{K}_1^2 \,\leq\, C'\,\varphi_x(h_r)$ which implies that $E\,Q_3^2 \,\leq\, C\,n^{-1}\,\varphi_x(h_r)^{-1}$ and it comes

$$Q_3 \,=\, O_P\left(1\,/\,\sqrt{n\,\varphi_x(h_r)}\right). \tag{4.16}$$

Now, thanks to (4.13)-(4.16), the proof of Theorem 1 is achieved as soon as Lemma 3 is established.

Proof of Lemma 3. It is easy to see that $\|x - X_1\| \,\leq\, h_r + \|\widehat{X}_1 - X_1\|$ with $\|\widehat{X}_1 - X_1\| \,=\, O_P(a_n)$ thanks to Lemma 1. In addition, $a_n = o(h_r)$ since $a_n^{1-\gamma} = o(h_r)$ by involving (H5-ii). In order to use a reductio ad absurdum, we assume that

$$\begin{aligned}
& \forall C > 0,\, \forall n_0,\, \exists n \geq n_0,\, P\left(\|x - X_1\|_2 \,<\, C\,h_r\right) \,<\, 1 \\
\Leftrightarrow\;\; & \forall C > 0,\, \forall n_0,\, \exists n \geq n_0,\, \exists \varepsilon_C,\, P\left(\|x - X_1\|_2 \,>\, C\,h_r\right) \,=\, \varepsilon_C \\
\Rightarrow\;\; & \forall n_0,\, \exists n \geq n_0,\, 0 < \varepsilon_C \leq P\left(\|x - X_1\|_2 \,>\, 2\,h_r\right) \leq P\left(\|\widehat{X}_1 - X_1\|_2 \,>\, h_r\right).
\end{aligned}$$

As $P\left(\|\widehat{X}_1 - X_1\|_2 \,>\, h_r\right) \leq (a_n/h_r)^2$ and because $a_n = o(h_r)$, it holds that for any $\tau > 0$, $\exists n_\tau > 0$, $\forall n > n_\tau$, $(a_n/h_r)^2 < \tau^2$. Now, setting $\tau = \sqrt{\varepsilon_C/2}$ and $n_0 = n_\tau$ allows to get the contradiction $0 \,<\, \varepsilon_C \,<\, \varepsilon_C/2$, which ends the proof of this Lemma.

Chapitre 5

Le traitement de données fonctionnelles en pratique

Ce chapitre est consacré à la mise en pratique des méthodes présentées dans ce manuscrit à l'aide du logiciel statistique R [170]. Nous proposons le développement de nouvelles méthodes permettant l'étude de données fonctionnelles. Les données étudiées, les codes R utilisés ainsi que les fichiers d'aide associés sont accessibles depuis le site de DYNAFOR : https://dynafor.toulouse.inra.fr/dynafornet/index.php/fre/Collaborateurs/Doctorants/Zullo. Trois méthodes fonctionnelles ont été développées et adaptées à l'étude de données hyperspectrales : une réécriture des méthodes non-paramétriques fonctionnelles développées par F. Ferraty et P. Vieu [87] visant à en réduire le temps de calcul, une extension fonctionnelle du modèle multinomial logistique au cadre de l'étude de données fonctionnelles, ainsi que la procédure d'implémentation de l'opérateur de régression non-paramétrique fonctionnel dans le cadre de l'étude de données bruitées. Une mise en pratique est proposée par l'écriture de codes R pour l'application de chacune de ces trois implémentations sur une partie des données hyperspectrales étudiées au cours de cette thèse. L'accent y est mis sur les protocoles expérimentaux utilisés ainsi que les résultats obtenus à chaque étape.

5.1 Optimisation du codage des méthodes non-paramétriques fonctionnelles

Les codes R développés pour l'implémentation des méthodes non-paramétriques fonctionnelles sont disponibles à l'adresse http://www.math.univ-toulouse.fr/~ferraty/SOFTWARES/NPFDA/npfda-routinesR.txt. La fonction R *funopadi.knn.lcv*, développée par F. Ferraty et P. Vieu [87], permet l'application de l'estimateur non-paramétrique fonctionnel pour la classification supervisée de courbes. L'écriture proposée ne permet cependant pas l'étude de gros volumes de données sans impliquer un coût élevé en termes de temps de calcul. Une optimisation de cette implémentation a été réalisée par l'écriture d'une fonction R *Funopadi_classif* pour la classification supervisée de données fonctionnelles. Ces deux implémentations diffèrent en plusieurs points. La principale amélioration apportée par notre écriture réside dans la suppression d'un maximum de boucles en privilégiant le calcul matriciel, permettant un gain de temps important mais au prix de l'occupation d'un plus grand volume de mémoire. L'utilisateur peut toutefois opérer un éventuel découpage des matrices d'apprentissage et de test en plusieurs sous-matrices par l'intermédiaire de paramètres de la fonction, permettant ainsi de trouver un compromis entre temps de calcul et mémoire nécessaire si les données sont trop volumineuses. Dans l'implé-

mentation originale, le choix du paramètre h a été remplacé par le choix local d'un paramètre équivalent des k plus proches voisins, pouvant varier d'une courbe à l'autre. Dans notre implémentation, le choix du paramètre des k plus proches voisins a été choisi de manière globale par validation croisée. Ainsi, le calcul de l'estimateur non-paramétrique fonctionnel prend en compte pour chaque courbe le même nombre de courbes voisines. Un noyau quadratique a été arbitrairement utilisé du fait de l'influence négligeable de ce choix sur la convergence de l'estimateur [87]. Dans le cadre de l'étude de données hyperspectrales, MPLSR s'est imposé pour le choix de la pseudométrique, car elle est particulièrement adaptée au traitement de ce type de données [223]. Par rapport à l'écriture originale, notre implémentation permet en plus un choix automatique par validation croisée du paramètre q associé à cette pseudométrique (nombre de composantes MPLSR), ainsi que la prise en compte de bruit dans les classes de l'échantillon d'apprentissage (le cas échéant).

La fonction R *Funopadi_classif* s'utilise de la façon suivante :

```
Funopadi_classif(responses, data, learnsamp, nbcomp=c(5, 10), nbcross=5,
                 maxsplit=1, noise=FALSE, noisedlabels=NULL, Huge=FALSE)
```

La fonction R *Funopadi_classif* nécessite en entrée :

`responses`	Un vecteur contenant la classe correspondant à chaque observation.
`data`	Une matrice d'observations, où chaque ligne représente une observation et chaque colonne une variable.
`learnsamp`	Une matrice d'entiers indexant les échantillons d'apprentissage (numéros de lignes de la matrice `data`), où chaque ligne conduit à une nouvelle répétition indépendante de la méthode. Tous les autres indices sont automatiquement affiliés aux échantillons test.
`nbcomp`	Un vecteur contenant différents nombres de composantes des moindres carrés partiels (PLS). Valeur par défaut : `c(5, 10)`.
`nbcross`	Nombre de parties souhaitées pour le calcul de validation croisée. Valeur par défaut : `5`.
`maxsplit`	Nombre de sous-échantillons à créer à partir de l'échantillon test. Paramètre purement calculatoire. Valeur par défaut : `1`.
`noise`	Paramètre booléen : s'il vaut `TRUE`, les classes d'apprentissage sont remplacées par les valeurs fournies dans `noisedlabels`. Valeur par défaut : `FALSE`.
`noisedlabels`	Nécessaire uniquement lorsque `noise=TRUE`. Une matrice contenant des valeurs de classes modifiées (bruitées), où chaque ligne conduit à une nouvelle répétition indépendante de la méthode. Valeur par défaut : `NULL`.
`Huge`	Paramètre booléen : s'il vaut `TRUE`, réduit l'espace mémoire nécessaire au calcul de la fonction mais augmente le temps de calcul en contrepartie. Paramètre purement calculatoire. Valeur par défaut : `FALSE`.

Nous conseillons de choisir `Huge=TRUE` si $\left(nblearn^2 \times nbclass/\texttt{nbcross}\right)$ est plus grand que 10^7, et une valeur pour le paramètre `maxsplit` au moins égale à $(nbclass \times nblearn \times nbtest/10^9)$, où $nbclass$ est le nombre total de classes dans les données, $nblearn$ and $nbtest$ sont respectivement le nombre d'observations utilisées pour construire chaque échantillon d'apprentissage et chaque échantillon test. Les deux paramètres `Huge` et `maxsplit` ne sont que purement calculatoires : ils n'ont aucune influence sur les paramètres obtenus en sortie. La longueur du vecteur `responses` et le nombre de lignes de la matrice `data` doivent correspondre. Si `noise=TRUE`, les dimensions des matrices `learnsamp` et `noisedlabels` doivent correspondre. Plusieurs répétitions de la méthode peuvent être calculées en un seul appel de la fonction.

La fonction R *Funopadi_classif* retourne en sortie une liste contenant :

`learnerror`	Un vecteur contenant les taux d'erreur de classification sur les échantillons d'apprentissage, une valeur pour chaque répétition.
`testerror`	Un vecteur contenant les taux d'erreur de classification sur les échantillons test, une valeur pour chaque répétition.
`confusion`	Une matrice de confusion en trois dimensions, où la première dimension correspond aux répétitions la deuxième dimension représente les classes prédites et la troisième dimension représente les classes observées.
`meanconfusion`	Matrice de confusion moyennée sur toutes les répétitions, où les lignes représentent les classes prédites et les colonnes représentent les classes observées.
`optnbcomps`	Un vecteur de paramètres pris dans `optnbcomp` optimisés par validation croisée, une valeur pour chaque répétition.
`optknearests`	Un vecteur de paramètres pris dans `optknearest` optimisés par validation croisée, une valeur pour chaque répétition.

Les codes R permettant l'implémentation de la fonction *Funopadi_classif* sont disponibles à l'adresse https://dynafor.toulouse.inra.fr/data/public/zullo/Funopadi_classif.R et le fichier d'aide associé «Funopadi_help.pdf» est disponible à l'adresse https://dynafor.toulouse.inra.fr/data/public/zullo/Funopadi_help.pdf.

Afin de mieux appréhender l'implémentation de la fonction R *Funopadi_classif*, nous proposons maintenant de présenter un cas concret d'application de cette fonction sur un exemple développé dans l'article [222] (chapitre 2 partie 2 de cette thèse). Le code R qui suit permet l'application de la méthode non-paramétrique fonctionnelle sur les données *AISA* pour 50 répétitions indépendantes du protocole expérimental suivant :

- 30 pixels d'apprentissage par classe,
- 30% de bruit dans les classes d'apprentissage (bruitage aléatoire uniforme sur l'ensemble des classes, voir chapitre 2 partie 2),
- Un choix du nombre de composantes MPLSR dans l'ensemble $\{2, 4, \ldots, 20\}$,
- Une optimisation simultanée du nombre de composantes MPLSR et du paramètre des k plus proches voisins par une validation croisée à 5 parties.

Dans ce cadre, la phase d'apprentissage est donc opérée en utilisant les classes bruitées, tandis que la phase de test compare les classes prédites avec les classes réellement observées (non bruitées).

Les codes R proposés dans la suite de cette section sont disponibles à l'adresse https://dynafor.toulouse.inra.fr/data/public/zullo/Code_application_Funopadi_classif.R.

Dans le logiciel R, commençons par charger uniquement les deux fichiers contenant les données brutes *AISA* (temps de chargement approximatif : 2 minutes) :

```
# Code R
temp <- tempfile()
download.file("https://dynafor.toulouse.inra.fr/data/public/zullo/AISA%20dataset
                                                                    .zip", temp)
Reflectances <- as.matrix(read.table(unz(temp, "x_aisa.txt")))
Cover_type <- unlist(read.table(unz(temp, "y_aisa.txt")))
unlink(temp)
# Fin code R
```

Les données étudiées dans l'article [222] (chapitre 2 partie 2 de cette thèse) ne sont en réalité qu'une partie des données *AISA*. En effet, en raison du très grand volume de ces données (362227 pixels pour 252 bandes spectrales), un sous-échantillonnage aléatoire représentatif des données (94232 pixels) a été opéré suivant les classes de la manière suivante :

- Tous les pixels pour les classes qui en contiennent moins de 10000,
- Un tiers des pixels pour les classes qui en contiennent entre 10000 et 50000,
- Un dizième des pixels pour les classes qui en contienent plus de 50000.

Le code R suivant réalise cet échantillonnage (le processus aléatoire est fixé afin de toujours obtenir les mêmes résultats) :

```
# Code R
subsample_sizes <- c(5885, 9308, 10110, 3442, 3581, 12602, 3754, 3804, 2094,
                                        8846, 4772, 6151, 3417, 2801, 13665)
subsample_index <- NULL
classes <- unique(Cover_type)
set.seed(242)
for (i in classes) {subsample_index <- c(subsample_index, sample(which(Cover_
                                                  type==i), subsample_sizes[i]))}
Reflectances_subsample <- Reflectances[sort(subsample_index), ]
Cover_type_subsample <- Cover_type[sort(subsample_index)]
# Fin code R
```

Un tirage aléatoire uniforme de 30 échantillons par classe est réalisé 50 fois indépendamment :

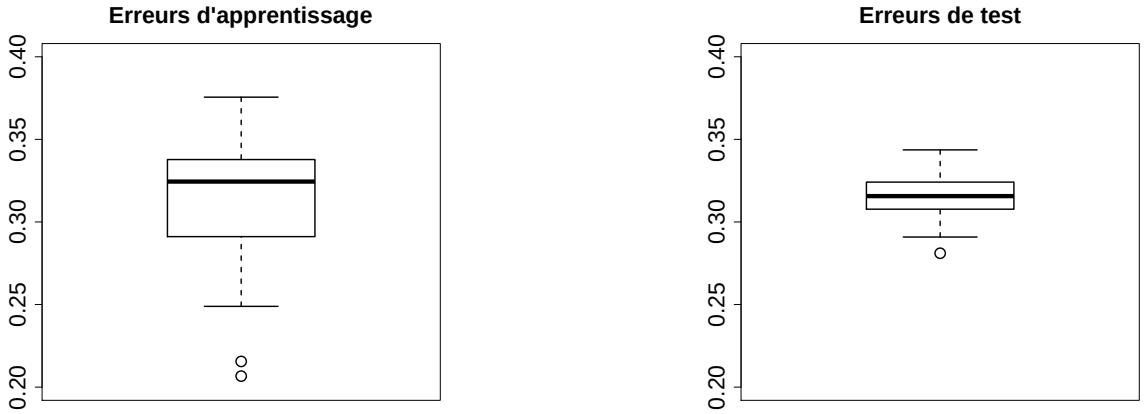
```
# Code R
learning <- NULL
for (i in classes) {learning <- cbind(learning, t(matrix(replicate(50, sample(
                                which(Cover_type_subsample==i), 30)), 30, 50))))}
# Fin code R
```

Il faut ensuite simuler les réponses bruitées pour les 50 échantillons à l'aide d'une loi multinomiale :

```
# Code R
nbclass <- length(classes)
nu <- 0.30
probas <- matrix(nu/(nbclass-1), nbclass, nbclass)
diag(probas) <- 1-nu
noised_Cover <- matrix(unlist(lapply(1:nbclass, function(x){classes[max.col(t(
                                rmultinom(1500, 1, probas[x, ])))]})), 50, 450)
# Fin code R
```

On charge la fonction *Funopadi_classif* ainsi que les fonctions nécessaires à son utilisation :

```
# Code R
file.lines <- scan("https://dynafor.toulouse.inra.fr/data/public/zullo/Funopadi
                    _classif.R", what=character(), skip=0, nlines=242, sep='\n')
file.lines.collapsed <- paste(file.lines, collapse='\n')
source(textConnection(file.lines.collapsed))
# Fin code R
```



(a) (b)

FIGURE 5.1 – Distributions des taux d'erreurs de classification sur l'échantillon d'apprentissage (a) et l'échantillon test (b) pour 50 répétitions de la méthode non-paramétrique fonctionnelle appliquée aux données *AISA* sous-échantillonnées. Les erreurs sont calculées en comparant les classes prédites avec les classes réellement observées (non bruitées), alors que le modèle est construit à partir des classes bruitées.

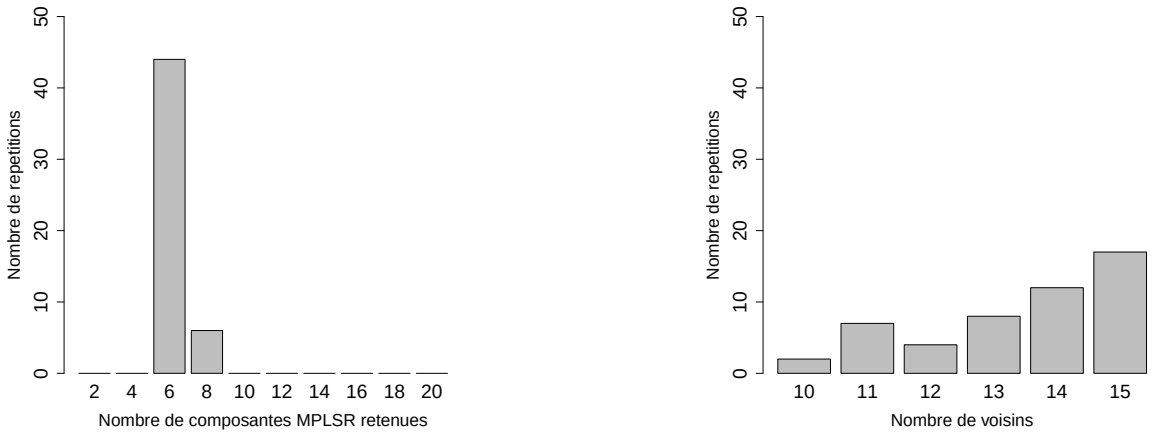
Nous pouvons maintenant appliquer la méthode non-paramétrique fonctionnelle de classification supervisée aux données *AISA* sous-échantillonnées selon le protocole défini précédemment (temps de calcul approximatif : 76 minutes) :

```
# Code R
Classification_results <- Funopadi_classif(Cover_type_subsample, Reflectances_
                                subsample, learning, nbcomp=2*(1:10), maxsplit
                                   =10, noise=TRUE, noisedlabels=noised_Cover)
# Fin code R
```

Ainsi, un seul appel de la fonction *Funopadi_classif* est nécessaire pour répéter l'application de la méthode pour plusieurs découpages d'échantillons apprentissage/test (ici 50 fois). Chaque échantillon test a été scindé en 10 parties de même taille afin d'éviter un encombrement trop important de la mémoire.

À partir des résultats obtenus, on peut par exemple afficher les erreurs moyennes et représenter sous forme de diagrammes en boîte (figure 5.1) les distributions respectives des erreurs sur les 50 échantillons d'apprentissage et les 50 échantillons test :

```
# Code R
print(paste("L'erreur moyenne sur l'echantillon d'apprentissage pour les 50
 repetitions vaut environ ", round(mean(Classification_results$learnerror),
 3), " soit ", 100*round(mean(Classification_results$learnerror), 3), "%"))
boxplot(Classification_results$learnerror, ylab="Erreurs d'apprentissage",
                ylim=c(0.2,0.4), cex=2, cex.lab=1.4, cex.axis=1.4, lwd=2)
x11()
print(paste("L'erreur moyenne sur l'echantillon test pour les 50 repetitions
           vaut environ ", round(mean(Classification_results$testerror), 3),
        " soit ",100*round(mean(Classification_results$testerror), 3), "%"))
```



(a) (b)

FIGURE 5.2 – Distribution des valeurs optimales du nombre de composantes MPLSR retenues (a) et du paramètre des k plus proches voisins (b) (Données *AISA*).

```
boxplot(Classification_results$testerror, ylab="Erreurs de test", ylim=c(0.2,0.4),
                                          cex=2, cex.lab=1.4, cex.axis=1.4, lwd=2)
# Fin code R
```

On obtient ici des taux d'erreur moyens de 31,4% pour l'échantillon d'apprentissage et 31,6% pour l'échantillon test. On constate (voir figure 5.1) que l'erreur moyenne reste à peu près la même entre échantillons d'apprentissage et échantillons test, mais que l'écart-type des erreurs est environ 3 fois plus important pour les échantillons d'apprentissage. Cette différence peut s'expliquer par le fait que les échantillons d'apprentissage ont une taille plus de 200 fois inférieure à celle des échantillons test, rendant les calculs d'erreurs moins stables.

Nous pouvons également visualiser la matrice de confusion moyennée (arrondie à l'unité) sur les 50 répétitions :

```
# Code R
print(round(Classification_results$meanconfusion, 0))
# Fin code R
```

Une matrice de confusion est un tableau croisé indiquant le nombre de pixels associés à chaque combinaison classe observée/classe prédite par la méthode. Cette matrice nous permet de constater que certaines confusions entre classes sont plus marquées que d'autres. Notamment, de nombreux pixels de la classe 1 («Luzerne») ont été prédits dans les classes 4, 14 et 15 (en moyenne plus de 3000 pixels pour «Jachère verte 2», plus de 1400 pour «Orge d'hiver» et plus de 1000 pour «Blé»). On notera également qu'en moyenne, plus de 1600 pixels de la classe 3 («Jachère verte 1») ont été prédits dans la classe 11 («Roseau»), et plus de 2200 pixels de la classe 13 («Eau») ont été prédits dans la classe 14 («Orge d'hiver»).

Intéressons nous maintenant aux moyennes et aux distributions des valeurs optimales du nombre de composantes MPLSR retenues et du paramètre des k plus proches voisins pour les 50 répétitions à l'aide de diagrammes en bâtons (figure 5.2) :

```
# Code R
print(paste("Le nombre moyen de composantes MPLSR retenues pour les 50 repetitions
```

```
                        est ", round(mean(Classification_results$optnbcomps), 0)))
barplot(table(factor(Classification_results$optnbcomps, levels=2*(1:10))), ylim=
              c(0,50), xlab="Nombre de composantes MPLSR retenues", ylab="Nombre
                    de repetitions", cex.lab=1.5, cex.axis=1.65, cex.names=1.65)
x11()
print(paste("La valeur moyenne du parametre des k plus proches voisins pour les
    50 repetitions est ", round(mean(Classification_results$optknearests), 0)))
barplot(table(factor(Classification_results$optknearests, levels=10:15)), ylim=
                              c(0,50), xlab="Nombre de voisins", ylab="Nombre de
                      repetitions", cex.lab=1.5, cex.axis=1.65, cex.names=1.65)
# Fin code R
```

Le graphique 5.2 (a) nous montre que le choix du nombre de composantes MPLSR retenues est stable (sur 50 répétitions, la valeur 6 a été sélectionnée 44 fois et la valeur 8 a été sélectionnée 6 fois). L'implémentation proposée calcule automatiquement un ensemble de valeurs candidates pour le paramètre des k plus proches voisins à partir du nombre total de classes et de la taille de l'échantillon d'apprentissage. Dans le cadre du protocole utilisé, cette implémentation n'autorise que les 6 valeurs 10, 11, 12, 13, 14 et 15. Le graphique 5.2 (b) peut nous laisser penser que lorsque le modèle est construit avec 30 échantillons par classe, la valeur optimale recherchée est possiblement supérieure à 15. Notre implémentation pourrait donc être améliorée en élargissant le choix du paramètre des k plus proches voisins à un intervalle de valeurs plus large, voire de laisser ce choix à l'utilisateur, au prix cependant d'un coût plus élevé en termes de temps de calcul.

La fonction *Funopadi_classif* permet donc d'appliquer la méthode non-paramétrique fonctionnelle de classification supervisée à un gros volume de données en un temps beaucoup plus raisonnable que l'approche originale de F. Ferraty et P. Vieu [87]. De plus, la capacité de répéter plusieurs fois l'application de cette méthode en un seul appel de la fonction rend cette nouvelle implémentation plus simple d'utilisation. Par ailleurs, le choix par l'utilisateur de divers modèles de bruit est facilité par leur application en amont et l'intégration directe des classes d'apprentissage bruitées dans l'appel de la fonction.

5.2 Extension du modèle multinomial logistique au cadre fonctionnel

Le modèle multinomial logistique est un cas particulier de modèle linéaire généralisé [152] pour la classification de données multivariées, extension du modèle binomial logistique à un nombre de classes supérieur à deux. Cette méthode n'est cependant pas adaptée à la classification de données fonctionnelles, comme en témoignent les résultats obtenus sur les données hyperspectrales *MADONNA* et *University of Pavia* [223]. Ainsi, l'écriture d'une adaptation fonctionnelle du modèle multinomial logistique apparaît nécessaire. Pour ce faire, nous avons choisi d'opérer une décomposition des données fonctionnelles dans une base de fonctions splines cubiques uniformes [14]. Les splines sont des fonctions continues, polynomiales par morceaux. Une spline est dite uniforme si tous ses morceaux sont des polynomes de même degré. Les points en lesquels ces morceaux se joignent sont appelés nœuds. La décomposition de données fonctionnelles dans une base de fonctions splines opère indirectement un lissage des données plus ou moins important selon le nombre de nœuds choisi (le lissage des données est d'autant plus marqué que le nombre de nœuds est faible). Cette décomposition transforme ainsi un espace fonctionnel en un espace multidimensionnel dont la dimension est égale au nombre de fonctions

de la base de splines considérée, dépendant du degré et du nombre de nœuds de ces fonctions. Ainsi, le modèle multinomial logistique devient applicable sur les données projetées en prenant les coefficients de la décomposition des données fonctionnelles dans la base de fonctions splines. La fonction R *Multifunc_classif*, écrite selon ce principe, permet l'application du modèle multinomial logistique à des données fonctionnelles. En pratique, le paramètre réglant le nombre de nœuds est automatiquement calculé par validation croisée. Cette fonction permet également la prise en compte de bruit dans les classes de l'échantillon d'apprentissage (le cas échéant), ainsi que le calcul en une seule commande de répétitions de la méthode pour plusieurs échantillons d'apprentissage.

La fonction R *Multifunc_classif* s'utilise de la façon suivante :

```
Multifunc_classif(responses, data, learnsamp, domrange, nbcross=5, noise=FALSE,
                  noisedlabels=NULL)
```

L'utilisation de cette fonction requiert l'installation des packages R splines, nnet et fda.

La fonction R *Multifunc_classif* nécessite en entrée :

`responses`	Un vecteur contenant la classe correspondant à chaque observation.
`data`	Une matrice d'observations, où chaque ligne représente une observation et chaque colonne une variable.
`learnsamp`	Une matrice d'entiers indexant les échantillons d'apprentissage (numéros de lignes de la matrice `data`), où chaque ligne conduit à une nouvelle répétition indépendante de la méthode. Tous les autres indices sont automatiquement affiliés aux échantillons test.
`domrange`	Un vecteur de taille 2 contenant les valeurs extrêmes (minimum et maximum) du domaine de définition des observations fonctionnelles.
`nbcross`	Nombre de parties souhaitées pour le calcul de validation croisée. Valeur par défaut : `5`.
`noise`	Paramètre booléen : s'il vaut `TRUE`, les classes d'apprentissage sont remplacées par les valeurs fournies dans `noisedlabels`. Valeur par défaut : `FALSE`.
`noisedlabels`	Nécessaire uniquement lorsque `noise=TRUE`. Une matrice contenant des valeurs de classes modifiées (bruitées), où chaque ligne conduit à une nouvelle répétition indépendante de la méthode. Valeur par défaut : `NULL`.

Les données sont décomposées sur une base de fontion splines dont le paramètre `nknot` (nombre de nœuds) est automatiquement optimisé par validation croisée. La longueur du vecteur `responses` et le nombre de lignes de la matrice `data` doivent correspondre. Si `noise=TRUE`, les dimensions des matrices `learnsamp` et `noisedlabels` doivent correspondre. Plusieurs répétitions de la méthode peuvent être calculées en un seul appel de la fonction.

La fonction R *Multifunc_classif* retourne en sortie une liste contenant :

`learnerror`	Un vecteur contenant les taux d'erreur de classification sur les échantillons d'apprentissage, une valeur pour chaque répétition.
`testerror`	Un vecteur contenant les taux d'erreur de classification sur les échantillons test, une valeur pour chaque répétition.
`confusion`	Une matrice de confusion en trois dimensions, où la première dimension correspond aux répétitions la deuxième dimension représente les classes prédites et la troisième dimension représente les classes observées.
`meanconfusion`	Matrice de confusion moyennée sur toutes les répétitions, où les lignes représentent les classes prédites et les colonnes représentent les classes observées.
`nknotopt`	Un vecteur de paramètres pris dans `nknot` optimisés par validation croisée, une valeur pour chaque répétition.

Les codes R permettant l'implémentation de la fonction *Multifunc_classif* sont disponibles à l'adresse https://dynafor.toulouse.inra.fr/data/public/zullo/Multifunc_classif.R et le fichier d'aide associé «Multifunc_help.pdf» est disponible à l'adresse https://dynafor.toulouse.inra.fr/data/public/zullo/Multifunc_help.pdf.

Afin de mieux appréhender l'implémentation de la fonction R *Multifunc_classif*, nous proposons maintenant de présenter un cas concret d'application de cette fonction sur un exemple développé dans l'article [222] (chapitre 2 partie 2 de cette thèse). Le code R qui suit permet l'application du modèle multinomial logistique fonctionnel sur les données *University of Pavia* pour 50 répétitions indépendantes du protocole expérimental suivant :

- 30 pixels d'apprentissage par classe,
- Un choix automatique du nombre de nœuds de la base de fonctions splines par une validation croisée à 5 parties.

Les codes R proposés dans la suite de cette section sont disponibles à l'adresse https://dynafor.toulouse.inra.fr/data/public/zullo/Code_application_Multifunc_classif.R.

Dans le logiciel R, commençons par charger uniquement les trois fichiers contenant les données brutes *University of Pavia* :

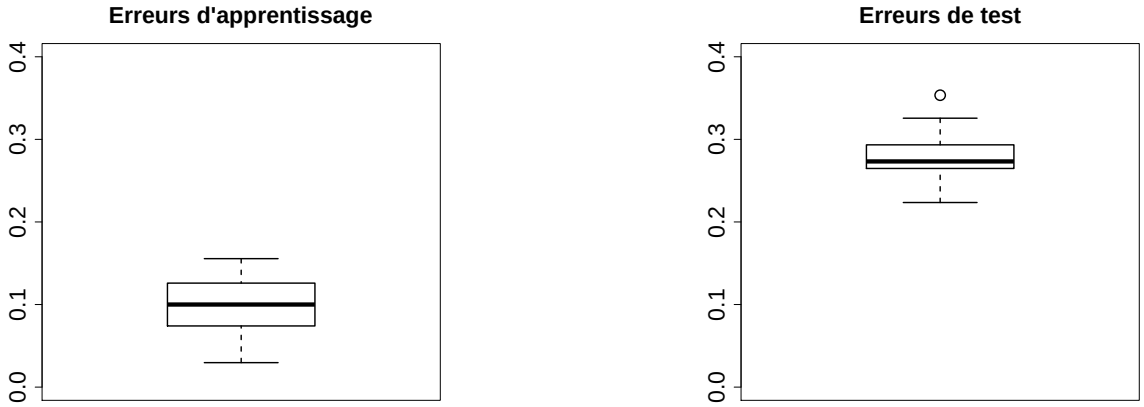
```
# Code R
temp <- tempfile()
download.file("https://dynafor.toulouse.inra.fr/data/public/zullo/University%20
                                                    of%20Pavia%20dataset.zip", temp)
Reflectances <- as.matrix(read.table(unz(temp, "university_of_pavia_x.txt")))
Material_type <- unlist(read.table(unz(temp, "university_of_pavia_y.txt")))
file.lines0 <- scan(unz(temp, "wave_uni.txt"), what=character(), sep='\n')
file.lines.collapsed0 <- paste(file.lines0, collapse='\n')
source(textConnection(file.lines.collapsed0))
unlink(temp)
# Fin code R
```

Un tirage aléatoire uniforme de 30 échantillons par classe est réalisé 50 fois indépendamment :

```
# Code R
classes <- unique(Material_type)
learning <- NULL
set.seed(105)
for (i in classes) {learning <- cbind(learning, t(matrix(replicate(50, sample(
                                        which(Material_type==i), 30)), 30, 50)))}
# Fin code R
```

On charge la fonction *Multifunc_classif* ainsi que les fonctions nécessaires à son utilisation :

```
# Code R (Necessite les packages splines, nnet et fda)
file.lines <- scan("https://dynafor.toulouse.inra.fr/data/public/zullo/Multifunc_
                      classif.R", what=character(), skip=0, nlines=105, sep='\n')
file.lines.collapsed <- paste(file.lines, collapse='\n')
source(textConnection(file.lines.collapsed))
# Fin code R
```



(a) (b)

FIGURE 5.3 – Distributions des erreurs sur l'échantillon d'apprentissage (a) et l'échantillon test (b) pour 50 répétitions de la méthode multinomiale logistique fonctionnelle appliquée aux données *University of Pavia*.

Nous pouvons maintenant appliquer la méthode multinomiale logistique fonctionnelle aux données *University of Pavia* selon le protocole défini précédemment (temps de calcul approximatif : 10 minutes) :

```
# Code R
Multifunc_results <- Multifunc_classif(Material_type, Reflectances, learning,
                                                        range(longueursdondes))
# Fin code R
```

Ainsi, un seul appel de la fonction *Multifunc_classif* est nécessaire pour répéter l'application de la méthode pour plusieurs découpages d'échantillons apprentissage/test (ici 50 fois).

À partir des résultats obtenus, on peut par exemple afficher les erreurs moyennes et représenter sous forme de diagrammes en boîte (figure 5.3) les distributions respectives des erreurs sur les 50 échantillons d'apprentissage et les 50 échantillons test :

```
# Code R
print(paste("L'erreur moyenne sur l'echantillon d'apprentissage pour les 50
        repetitions vaut environ ", round(mean(Multifunc_results$learnerror),
        3)," soit ", 100*round(mean(Multifunc_results$learnerror), 3), "%"))
boxplot(Multifunc_results$learnerror, main="Erreurs d'apprentissage", ylim=
                                c(0, 0.4), cex=2, cex.main=2, cex.axis=2, lwd=2)
x11()
print(paste("L'erreur moyenne sur l'echantillon test pour les 50 repetitions
                vaut environ ", round(mean(Multifunc_results$testerror), 3),
                " soit ", 100*round(mean(Multifunc_results$testerror), 3), "%"))
boxplot(Multifunc_results$testerror, main="Erreurs de test", ylim=c(0, 0.4),
                                      cex=2, cex.main=2, cex.axis=2, lwd=2)
# Fin code R
```

On obtient ici des taux d'erreur moyens de 9,8% pour l'échantillon d'apprentissage et 27,7% pour l'échantillon test. Cette erreur est donc en moyenne presque 3 fois plus élevée dans l'échan-

Classe observée	1	2	3	4	5	6	7	8	9
Classe prédite									
1	**4400**	70	140	1	13	98	1	206	21
2	15	**13082**	3	283	0	4	7	0	1094
3	503	127	**1473**	1	0	725	0	18	57
4	25	1422	5	**2570**	5	5	10	1	90
5	80	76	1	55	**888**	1	6	0	17
6	298	268	364	18	0	**2543**	11	9	238
7	86	85	2	29	9	2	**1270**	2	15
8	1130	5	49	2	1	109	8	**1060**	6
9	65	3484	33	75	0	165	1	3	**3461**

TABLE 5.1 – Nombre moyen de pixels (arrondi à l'unité) associé à chaque couple classe observée/classe prédite par la méthode multinomiale logistique fonctionnelle pour 50 répétitions (Données *University of Pavia*). Les valeurs en gras correspondent aux pixels bien classés.

tillon test. Cette dégradation de la qualité de prédiction du modèle laisse ainsi apparaître un phénomène de surapprentissage, induisant une perte de capacité de prédiction de la méthode. Cette perte de généralisabilité est d'autant plus accentuée lorsque l'échantillon d'apprentissage est de petite taille.

Visualisons maintenant la matrice de confusion moyennée (arrondie à l'unité) sur les 50 répétitions (Table 5.1) :

```
# Code R
print(round(Multifunc_results$meanconfusion, 0))
# Fin code R
```

La table 5.1 nous permet de constater que certaines confusions entre classes sont plus marquées que d'autres. En particulier, les classes 2 («Prairie») et 9 («Ombre») sont particulièrement impactées par une confusion réciproque (plus de 4500 pixels en moyenne). La classe 2 («Prairie») compte également plus de 1400 pixels prédits dans la classe 4 («Arbre»), et plus de 1100 pixels de la classe 1 («Asphalte») ont été prédits dans la classe 8 («Brique»).

Visualisons la différence entre quatre courbes brutes aléatoirement choisies et la reconstruction de leur version décomposée dans la base de fonctions splines cubiques avec un nombre de nœuds égal à 3 (figure 5.4) :

```
# Code R
set.seed(13)
Four_curves <- Reflectances[sample(nrow(Reflectances), 4), ]
p <- ncol(Four_curves)
a <- min(longueursdondes)
b <- max(longueursdondes)
x <- seq(a, b, length=p)
order.Bspline <- 4
nknot <- 3
Knot <- seq(a, b, length=nknot+2)[-c(1, nknot+2)]
delta <- sort(c(rep(c(a, b), order.Bspline), Knot))
Bspline <- splineDesign(delta, x, order.Bspline)
coefs <- t(symsolve(t(Bspline)%*%Bspline, t(Bspline)%*%t(Four_curves)))
Smoothed_curves <- coefs%*%t(Bspline)
```

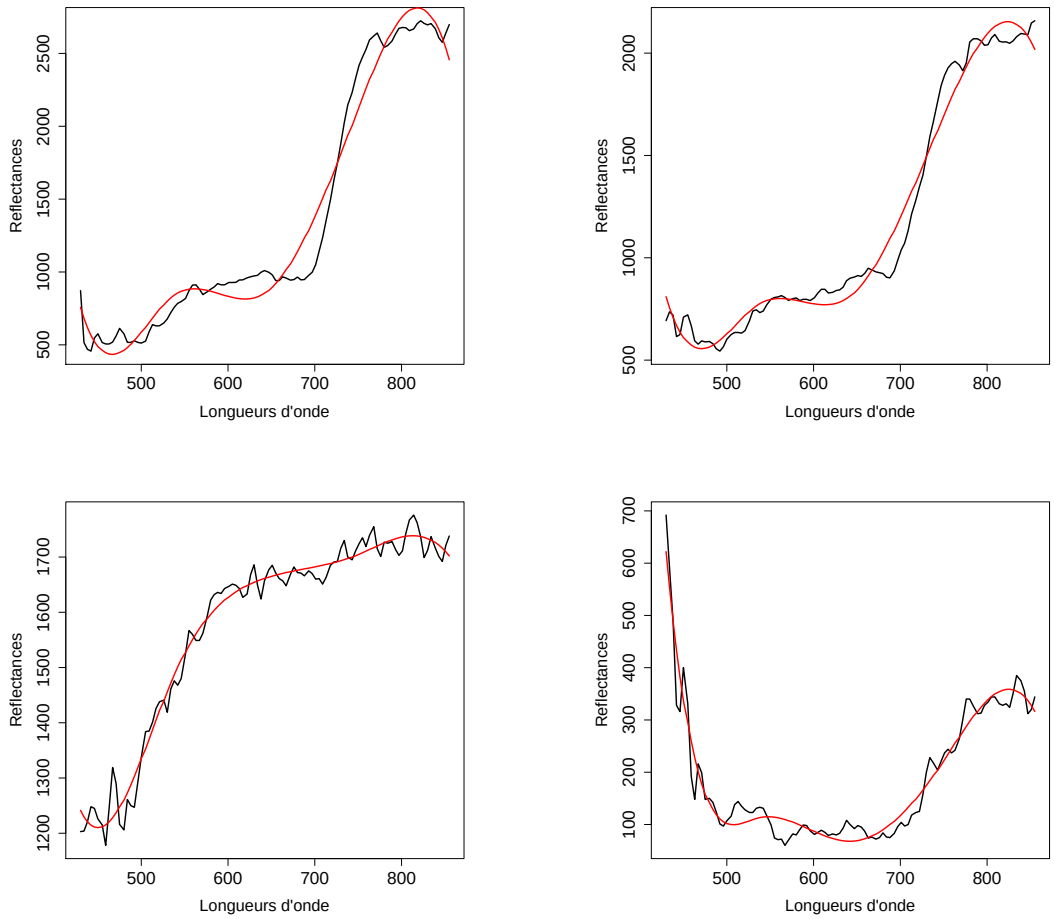


FIGURE 5.4 – Quatre courbes brutes des données *University of Pavia* (en noir) et leur version lissée par la décomposition dans une base de fonction splines à 3 nœuds (en rouge).

```
par(mfrow=c(2,2))
for (i in 1:4) {plot(longueursdondes, Four_curves[i, ], type="l", xlab="Longueurs
                d'onde", ylab="Reflectances", lwd=2, cex.lab=1.4, cex.axis=1.5);
                lines(longueursdondes, Smoothed_curves[i, ], col="red", lwd=2)}
# Fin code R
```

On constate, d'après la figure 5.4, que la décomposition des données dans une base de fonctions splines opère un lissage de ces données (plus ou moins fort selon le nombre de nœuds). Ainsi, le modèle multinomial logistique n'est plus réellement appliqué aux données brutes mais aux coefficients de cette décomposition (équivalent à une application sur les données ainsi lissées).

La fonction *Multifunc_classif* permet donc d'appliquer le modèle multinomial logistique pour l'étude de données fonctionnelles. De plus, la capacité de répéter plusieurs fois l'application de cette méthode en un seul appel de la fonction rend cette nouvelle implémentation plus simple d'utilisation. Par ailleurs, le choix par l'utilisateur de divers modèles de bruit est facilité par leur application en amont et l'intégration directe des classes d'apprentissage bruitées dans l'appel de la fonction.

5.3 Estimateur non-paramétrique fonctionnel et lissage des données

Nous avons implémenté deux estimateurs non-paramétriques fonctionnels opérant un prétraitement de lissage des données : *Nested_Funopare* pour la régression et *Nested_Funopadi* pour la classification supervisée, respectivement adaptés des fonctions R *funopare.knn.lcv* et *funopadi.knn.lcv* développés par F. Ferraty et P. Vieu [87] (dont les codes R sont disponibles à l'adresse http://www.math.univ-toulouse.fr/~ferraty/SOFTWARES/NPFDA/npfda-routinesR.txt). Tandis que l'implémentation originale impose pour le choix du paramètre des k plus proches voisins un ensemble de valeurs automatiquement défini en fonction de la taille de l'échantillon d'apprentissage, notre écriture permet un libre choix de cet ensemble par l'utilisateur. En plus de ce paramètre, cette extension conduit au réglage d'un nouveau paramètre de lissage, représentant un coefficient de proportionnalité entre la taille de la fenêtre de lissage du noyau et l'écart-type inter-spectres en chaque point de discrétisation. Cinq pseudométriques standard sont mises à disposition avec ces fonctions. Trois d'entre elles ont déjà été présentées dans la section 1.1.4 de ce manuscrit : «deriv» (basée sur les dérivées successives du prédicteur fonctionnel), «pca» (basée sur une analyse fonctionnelle en composantes principales), et «mplsr» (basée sur une décomposition des moindres carrés partiels). Les deux autres, «fourier» et «hshift», sont respectivement basées sur une décomposition dans une base de Fourier et sur la prise en compte d'un éventuel déphasage des données fonctionnelles. Il est également possible d'implémenter d'autres pseudométriques afin d'adapter l'estimateur non-paramétrique à l'étude de tous types de données fonctionnelles (comme par exemple BAGIDIS [203]).

Les fonctions R *Nested_Funopare* et *Nested_Funopadi* s'utilisent de la façon suivante :

```
Nested_Funopare(Responses, CONTAMINATED_CURVES, learnsamp, semimetric, ...,
                Bwsmooth, Knn, Grid, kfold=5, adaptive=TRUE)
Nested_Funopadi(Classes, CONTAMINATED_CURVES, learnsamp, semimetric, ...,
                Bwsmooth, Knn, Grid, kfold=5, adaptive=TRUE)
```

Les fonctions R *Nested_Funopare* et *Nested_Funopadi* nécessitent en entrée :

`Responses`	Un vecteur contenant la valeur numérique (scalaire) correspondant à chaque observation (fonction *Nested_Funopare* uniquement).
`Classes`	Un vecteur contenant la classe correspondant à chaque observation (fonction *Nested_Funopadi* uniquement).
`CONTAMINATED_CURVES`	Une matrice d'observations bruitées, où chaque ligne représente une observation et chaque colonne une variable.
`learnsamp`	Un vecteur d'entiers indexant l'échantillon d'apprentissage (numéros de lignes de la matrice `CONTAMINATED_CURVES`). Tous les autres indices sont automatiquement affiliés à l'échantillon test.
`semimetric`	Une chaine de caractères définissant le choix de la pseudométrique. Les cinq valeurs possibles sont : `"deriv"` : pseudométrique basée sur la dérivation des observations fonctionnelles. `"fourier"` : pseudométrique basée sur une décomposition des observations fonctionnelles dans une base de Fourier. `"hshift"` : pseudométrique basée sur un recalage horizontal des données fonctionnelles. `"mplsr"` : pseudométrique basée sur une décomposition des moindres carrés partiels des observations fonctionnelles. `"pca"` : pseudométrique basée sur l'analyse fonctionelle en composantes principales.
`...`	Les valeurs des paramètres spécifiques de la pseudométrique choisie.
`Bwsmooth`	Un vecteur contenant différentes valeurs du paramètre de lissage à noyau quadratique.
`Knn`	Un vecteur contenant différentes valeurs du paramètre des k plus proches voisins de l'estimateur non-paramétrique fonctionnel.
`Grid`	Un vecteur contenant les valeurs des points de discrétisation des observations fonctionnelles.
`kfold`	Nombre de parties souhaitées pour le calcul de validation croisée. Valeur par défaut : `5`.
`adaptive`	Paramètre booléen : s'il vaut `TRUE`, le lissage à noyau opéré sera en chaque point de la discrétisation proportionnel à l'écart-type inter-spectres. Valeur par défaut : `TRUE`.

Ces deux fonctions ont été codées de façon à paralléliser les calculs de validation croisée. La seule différence entre leurs paramètres d'entrée réside dans la nature de la variable réponse. Si le problème statistique considéré est la régression (repectivement la classification supervisée), alors la variable réponse sera de nature scalaire (respectivement catégorielle) et on utilisera la fonction *Nested_Funopare* (respectivement *Nested_Funopadi*). Les listes de paramètres spécifiques nécéssaires au calcul de la pseudométrique choisie sont disponibles en ligne (voir https://dynafor.toulouse.inra.fr/data/public/zullo/Nested_Funopa.R). Si `adaptive=FALSE`, un lissage à noyau quadratique uniforme sera appliqué. Le nombre de lignes de la matrice `CONTAMINATED_CURVES` doit correspondre à la longueur du vecteur `Responses` (pour *Nested_Funopare*) ou du vecteur `Classes` (pour *Nested_Funopadi*). La longueur du vecteur `Grid` doit correspondre au nombre de colonnes de la matrice `CONTAMINATED_CURVES`.

La fonction R *Nested_Funopare* retourne en sortie une liste contenant :

`testerror`	La valeur de l'erreur relative de prédiction sur l'échantillon test.
`CVerror`	Une matrice contenant les erreurs relatives de validation croisée, où chaque ligne correspond à une valeur du vecteur `Knn` et chaque colonne à une valeur de `Bwsmooth`.
`Bwsmoothopt`	La valeur du paramètre de `Bwsmooth` optimisée par validation croisée.
`Knnopt`	La valeur du paramètre de `Knn` optimisée par validation croisée.

La fonction R *Nested_Funopadi* retourne en sortie une liste contenant :

`testerror`	Le taux d'erreur de classification sur l'échantillon test.
`CVerror`	Une matrice contenant les erreurs relatives de validation croisée, où chaque ligne correspond à une valeur du vecteur `Knn` et chaque colonne à une valeur de `Bwsmooth`.
`Bwsmoothopt`	La valeur du paramètre de `Bwsmooth` optimisée par validation croisée.
`Knnopt`	La valeur du paramètre de `Knn` optimisée par validation croisée.
`confusion`	Matrice de confusion, où les lignes représentent les classes prédites et les colonnes représentent les classes observées.

Les codes R permettant l'implémentation de ces deux fonctions sont disponibles à l'adresse https://dynafor.toulouse.inra.fr/data/public/zullo/Nested_Funopa.R et le fichier d'aide associé «Nested_Funopa_help.pdf» est disponible à l'adresse https://dynafor.toulouse.inra.fr/data/public/zullo/Nested_Funopa_help.pdf. Des codes R proposant une application de chacune de ces deux fonctions sur des données fonctionnelles sont disponibles à l'adresse https://dynafor.toulouse.inra.fr/data/public/zullo/Code_application_Nested_Funopa.R.

Chapitre 6

Discussion

Un spectre provient de l'observation de la reflectance en d points de mesure produisant d variables. De plus, la constitution d'échantillon d'apprentissage étant onéreuse (collection de spectres associée à une variable réponse d'intérêt comme par exemple le type de végétation), on se retrouve dans une situation où la taille d'échantillon n est relativement faible devant le nombre d de variables. Ce phénomène de «grande dimension» appelé «fléau de la dimension», est bien connu en statistique multivariée. Il s'agit réellement d'un fléau puisque plus d augmente devant n, plus les performances des méthodologies statistiques standard se dégradent. Et ceci est d'autant plus vrai dans le cadre hyperspectral que les progrès technologiques favorisent un accroissement de d. Or, nous avons vu que les spectres de reflectance intègrent dans leur dimension spectrale un continuum qui leur confère une nature fonctionnelle. Ainsi, un hyperspectre peut être modélisé par une fonction univariée de la longueur d'onde, sa représentation produisant une courbe. L'utilisation de méthodes fonctionnelles sur de telles données permet de prendre en compte des aspects fonctionnels tels que la continuité, l'ordre des bandes spectrales et de s'affranchir des fortes corrélations liées à la finesse de la grille de discrétisation. Nous avons aussi remarqué que, lorsqu'on dispose de données fonctionnelles contaminées, pour une taille d'échantillon fixée, plus la discrétisation est fine, meilleure sera la prédiction. Autrement dit, plus d est grand devant n, plus la méthode statistique fonctionnelle développée est performante. Ceci nous amène à deux importantes remarques :

1. Les évolutions technologiques de l'imagerie hyperspectrale qui conduisent à une augmentation de la finesse de la grille de discrétisation plaident en faveur de l'utilisation des méthodes statistiques fonctionnelles,
2. Contrairement à l'approche multivariée, le cadre «grande dimension» est une bénédiction pour les méthodes fonctionnelles.

On peut toutefois nuancer ces conclusions. Dans un cadre de classification supervisée de données hyperspectrales, les méthodes multivariées et les méthodes fonctionnelles fournissent des approches complémentaires. Tandis que les méthodes multivariées sont plus performantes dans les situations statistiques favorables (forte hétérogénéité des classes, échantillons de taille suffisante, absence de bruit dans les données), les méthodes fonctionnelles les surpassent lorsqu'une situation est plus défavorable (faible hétérogénéité des classes, données bruitées) et laisse apparaître le «fléau de la dimension» (échantillons de taille petite devant le nombre de variables). En particulier, les méthodes non-paramétriques fonctionnelles sont particulièrement adaptées à l'étude d'images hyperspectrales. Ces méthodes nécessitent cependant de définir une notion spécifique de mesure de proximité entre deux courbes tenant compte des caractéristiques plus ou moins fines des variables mises en jeu. En effet, l'utilisation d'une distance fonctionnelle standard dans le calcul de l'estimateur ne permet pas d'éviter le «fléau de la dimension». L'extension du choix de cette mesure à des pseudométriques permet ainsi d'opérer une réduction

de la dimension du problème (théoriquement infinie pour des données fonctionnelles). De plus, cette pseudométrique a une influence importante sur la qualité de prédiction de l'estimateur selon la nature des données considérées. Le choix de la pseudométrique mplsr, basée sur une décomposition des moindres carrés partiels, s'est révélé particulièrement pertinent dans le cadre de l'étude de données hyperspectrales par des méthodes non-paramétriques fonctionnelles. Cependant, la présence de bruit dans les données hyperspectrales nécessite le développement de méthodes fonctionnelles spécifiquement adaptées. C'est pourquoi l'implémentation d'un nouvel estimateur non-paramétrique fonctionnel comprenant une étape de lissage (i.e., débruitage) des données permet une amélioration significative des résultats obtenus sans lissage. D'un point de vue théorique, confirmé par la pratique, cette nouvelle méthode est d'autant moins impactée par le bruit que le nombre de points de discrétisation est important. Ainsi, l'amplification de la finesse de discrétisation des relevés hyperspectraux pourra potentiellement permettre d'obtenir une amélioration des résultats.

Les variables consécutives résultant de l'échantillonnage de données hyperspectrales apparaîssent fortement corrélées entre elles. Ainsi, la redondance de l'information contenue dans ces variables provoque une baisse plus ou moins importante de la capacité de prédiction de la plupart des méthodes multivariées standard. Certaines méthodes statistiques non fonctionnelles permettent malgré tout l'étude de données hyperspectrales. Devant le grand nombre de variables ainsi échantillonnées, des méthodes qui opèrent une réduction de la dimension du problème sont potentiellement capables de fortement diminuer cette redondance. Le développement de méthodes multivariées parcimonieuses peut donc s'avérer une piste pertinente dans l'étude d'images hyperspectrales. Dans ce cadre, les deux méthodes présentées dans le chapitre 3 de cette thèse : Most Predictive Design Points (MPDP) et Nonlinear Parsimonious Feature Selection (NPFS), réalisent une sélection des variables spectrales les plus prédictives selon un critère prédéfini. Ces méthodes parviennent à garder de bonnes capacités de généralisation en ne sélectionnant que quelques dizaines de variables parmi des milliers. De plus, les méthodes de sélection de variables permettent une plus grande interprétabilité des bandes spectrales sélectionnées que les méthodes d'extraction de caractéristiques. Cependant, le manque de stabilité des bandes spectrales sélectionnées dû aux fortes corrélations entre les variables nuit considérablement à cette interprétabilité. Ainsi, le développement de méthodes de sélection d'intervalles spectraux continus pourrait permettre de pallier ce problème tout en prenant en compte la nature fonctionnelle des données hyperspectrales.

Dans l'ensemble de ce travail de thèse, seule la dimension spectrale des données a été prise en compte dans la modélisation statistique. Ainsi, l'apport d'information supplémentaire (spatiale, temporelle, ...) pourrait permettre une amélioration substantielle de la capacité de prédiction des modèles. De nombreux travaux font par exemple état d'études d'images hyperspectrales aéroportées couplant la dimension spectrale avec la dimension spatiale par la modélisation de dépendances entre pixels proches [74, 194, 195, 140]. Cependant, les méthodes utilisées pour résoudre ces problèmes appréhendent la dimension spectrale uniquement dans un cadre multivarié. Ainsi, le développement de méthodes fonctionnelles pour l'étude spectrale/spatiale d'images hyperspectrales pourrait fournir une alternative intéressante. Dans le cadre du modèle non-paramétrique fonctionnel, l'estimateur de Nadaraya-Watson pourrait par exemple être adapté à la résolution de ce type de problèmes par l'utilisation d'un noyau bivarié (spectral/spatial) au lieu d'un noyau spectral univarié. On peut ainsi espérer une meilleure qualité de prédiction que pour la seule prise en compte de la dimension spectrale, malgré un coût plus élevé en termes de volume de données et de temps de calcul.

Par nature, les données temporelles sont un exemple de données fonctionnelles. Ce type de données a été étudié à l'aide de méthodes multivariées [208, 91, 3, 48], mais on ne trouve que peu d'applications de méthodes fonctionnelles. Ainsi, l'étude d'images à haute résolution temporelle

pourrait directement être investiguée à travers le prisme des méthodes fonctionnelles, de façon analogue à l'étude de données hyperspectrales.

Annexe A

Quelques jeux de données hyperspectraux

Les sections suivantes présentent les principaux jeux de données hyperspectraux utilisés au cours de cette thèse. Les trois premiers sont des données réelles obtenues à l'aide de capteurs hyperspectraux aéroportés. Ils ont en commun un nombre important de pixels référencés (plus de 30000 pixels), un nombre conséquent de bandes spectrales (plus d'une centaine), une couverture spectrale dans le visible et le début du proche infrarouge (400-1000 nm environ) et une résolution spectrale (longueur séparant deux bandes spectrales consécutives) de l'ordre de quelques nanomètres seulement. Leurs principales différences résident dans la nature des données et la résolution spatiale des capteurs utilisés. Le dernier jeu de données, basé sur un modèle de simulation biophysique, se distingue des trois autres de façon importante par un nombre plus réduit de courbes hyperspectrales, un nombre beaucoup plus important de bandes spectrales et une couverture spectrale plus étendue.

A.1 Les données *MADONNA*

Les données *MADONNA* sont composées de 32224 pixels sélectionnés dans trois images hyperspectrales, relevées par un capteur hyperspectral aéroporté de type HYSPEX sur le site de Villelongue dans les Pyrénées françaises. La figure A.1 présente la composition vraies couleurs d'une partie du site d'étude. Chaque spectre de réflectance de ces images a été décomposé selon 160 bandes spectrales équi-espacées dans la gamme visible/proche infrarouge (400-1000 nm), avec une résolution spectrale de 1,5 nm et une résolution spatiale de 50 cm. Parallèlement à la prise de ces images hyperspectrales, une campagne de terrain d'une semaine a été menée afin d'identifier les espèces arborées en présence. À chacun de ces pixels correspond l'une des 12 espèces arborées identifiées sur la zone d'étude. Les 32224 pixels de ces données se répartissent en 12 classes comme suit : Châtaignier (2855), Noyer (1016), Tilleul (3402), Frêne (4333), Érable (165), Chêne (10981), Fougère (1983), Noisetier (4122), Hêtre (42), Bouleau (468), Saule Marsault (485) et Robinier (2372). Ces pixels ne représentent qu'environ 1% de la surface totale de la zone. En effet, malgré la très bonne résolution du capteur, beaucoup de pixels sont composés d'un mélange de plusieurs espèces («mixels»). D'autres pixels contiennent des éléments parasites tels que les champs, les routes ou les bâtiments, inutiles vis-à-vis du projet. Le référencement de chaque pixel a été effectué manuellement à partir des relevés obtenus sur le terrain. La figure A.2 présente pour chacune des 12 espèces un échantillon de 10 hyperspectres (en noir), ainsi que le profil spectral moyen de la classe correspondante (en rouge).

FIGURE A.1 – Composition vraies couleurs d'une partie de la zone d'étude du site de Villelongue, Pyrénées. Certains pixels inclus dans les données ont été représentés en fausses couleurs selon l'espèce arborée correspondante.

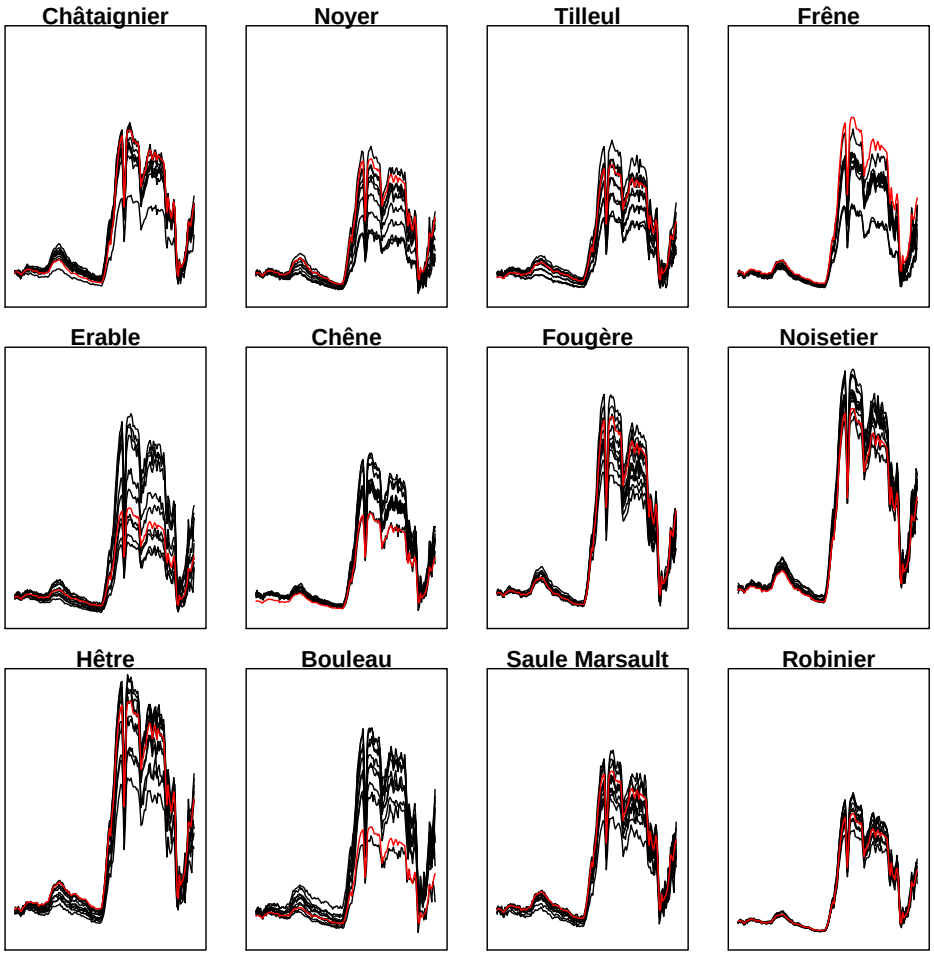


FIGURE A.2 – Profil spectral moyen (en rouge) de chacune des 12 espèces et 10 hyperspectres (en noir) aléatoirement choisis dans chacune des classes (données *MADONNA*).

A.2 Les données *University of Pavia*

Les données *University of Pavia* [74] ont été collectées à l'aide d'un capteur optique aéroporté de type ROSIS-03 sur le site de l'université de Pavie au nord de l'Italie. Ces données sont

(a) (b)

FIGURE A.3 – Composition vraies couleurs du site de l'université de Pavie, Italie (a) et représentation des classes (b).

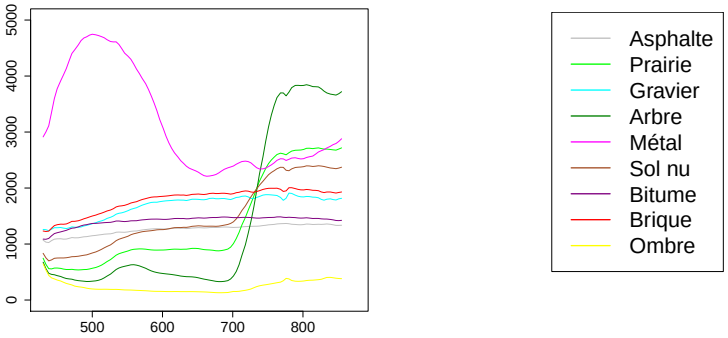
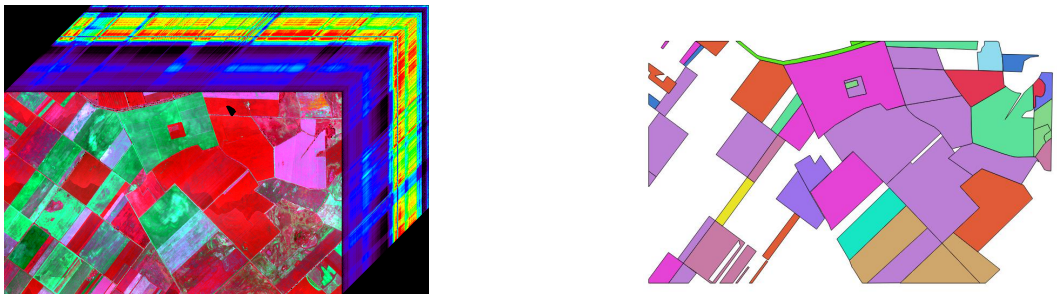


FIGURE A.4 – Profil spectral moyen de chacune des 9 classes (données *University of Pavia*).

constituées de 42776 pixels décomposés en 103 bandes spectrales, avec une couverture spectrale de 430 à 860 nm et une résolution spatiale de 1,3 m par pixel, pour un total de 9 classes incluant matériaux urbains − Asphalte (6631), Gravier (2099), Métal (1345), Bitume (1330), Brique (3682) −, végétation − Prairie (18649), Arbre (3064) −, Sol nu (5029) et Ombre (947). Contrairement aux données *MADONNA*, ces données présentent une importante hétérogénéité thématique, rendant la discrimination a priori plus aisée malgré une résolution spatiale un peu moins fine. La figure A.3 montre une composition vraies couleurs de ce site (a) ainsi qu'une représentation des classes (b). La figure A.4 représente le profil spectral moyen de chacune de ces classes. Ces données (fichier archive zip) sont disponibles à l'adresse : https://dynafor.toulouse.inra.fr/data/public/zullo/University%20of%20Pavia%20dataset.zip.

A.3 Les données *AISA*

Les données *AISA* [210] ont été récoltées à l'aide d'un capteur hyperspectral de type AISA Eagle dans une zone contenant des terres cultivables près de la ville de Heves en Hongrie, avec une résolution spatiale modifiée de 6 m par pixel (initialement 2 m par pixel). La figure A.5 montre une représentation de l'image hyperspectrale correspondante à ces données (a) ainsi



(a) (b)

FIGURE A.5 – Représentation de l'image hyperspectrale *AISA* (a) et représentation des classes (b).

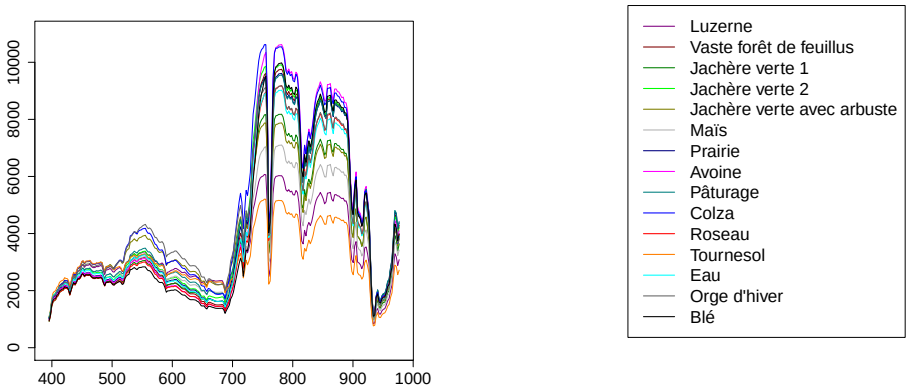


FIGURE A.6 – Profil spectral moyen de chacune des 15 classes (données *AISA*).

qu'une représentation de l'image selon la classe de chaque pixel (b). Au total, 362227 pixels ont été décomposés selon 252 bandes couvrant la zone spectrale 395-975 nm, et répartis en 15 classes de la façon suivante : Luzerne (17656), Vaste forêt de feuillus (9308), Jachère verte 1 (30331), Jachère verte 2 (3442), Jachère verte avec arbuste (10743), Maïs (37805), Prairie (3754), Avoine (11412), Pâturage (2094), Colza (26537), Roseau (4772), Tournesol (61508), Eau (3417), Orge d'hiver (2801) et Blé (136647). Le profil spectral moyen de chacune des 15 classes étudiées est présenté dans la figure A.6. Comme semble nous l'indiquer cette figure, la nature végétale commune à la grande majorité de ces 15 classes conduit à leur représentation par des profils spectraux similaires, à l'instar des données *MADONNA*. Cependant, la dégradation de la résolution spatiale (chaque pixel de la nouvelle résolution contient en réalité 9 pixels originaux) conduit à l'apparition de mixels (pixels contenant un mélange de plusieurs objets de natures différentes), compliquant encore davantage la classification de ces données. Ce jeu de données revêt donc un caractère particulièrement intéressant de par cette combinaison entre nature similaire des données et résolution spatiale dégradée, conduisant à l'étude de données potentiellement difficiles à classifier. Étant donné le très grand nombre de pixels disponibles (environ 10 fois plus que pour les données *MADONNA* ou *University of Pavia*), nous n'avons utilisé dans l'article [222] (chapitre 2 partie 2 de cette thèse) qu'un sous-échantillon représentatif de ce jeu de données. Ces données (fichier archive zip) sont disponibles à l'adresse :

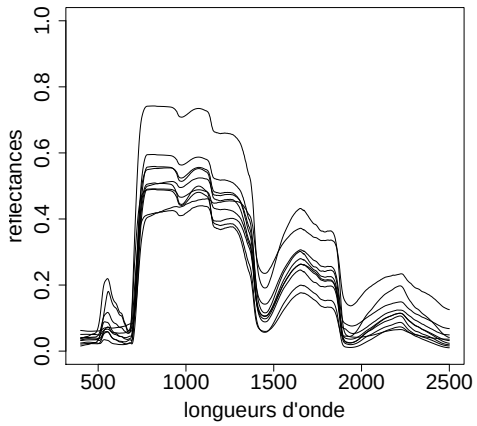


FIGURE A.7 – Quelques spectres simulés (données *PROSAIL*).

https://dynafor.toulouse.inra.fr/data/public/zullo/AISA%20dataset.zip.

A.4 Les données *PROSAIL*

Ces données proviennent d'une simulation de 5000 hyperspectres de réflectance de feuilles végétales opérée par le modèle biophysique *PROSAIL* [116] à partir de 5 variables biophysiques (la concentration en chlorophylle, la concentration en caroténoïdes, l'épaisseur en eau, la masse surfacique de matière sèche et la concentration en azote dans la structure de la feuille). Les hyperspectres ont été discrétisés selon 2101 variables uniformément réparties entre 400 et 2500 nm, soit 1 nm de résolution spectrale. La spécificité de ce jeu de données provient de sa nature. En effet, contrairement aux autres qui mettent en jeu des courbes issues de relevés de données réelles, ce jeu de données a été obtenu par simulations d'un modèle biophysique (non-statistique). Ainsi, malgré le caractère simulé de ces données, le lien statistique existant entre ces spectres et les variables biophysiques à partir desquels ils ont été construits n'est pas connu. Par ailleurs, il est également le seul à s'incrire dans un cadre de régression parmi les quatre jeux de données présentés dans cette annexe. Parmi les variables biophysiques mises en jeu, nous avons choisi de porter notre intérêt sur la variable «concentration en chlorophylle». Quelques profils spectraux simulés sont présentés dans la figure A.7.

Bibliographie

[1] A. Agarwal, T. El-Ghazawi, H. El-Askary, and J. Le Moigne. Efficient hierarchical-PCA dimension reduction for hyperspectral imagery. In *IEEE International Symposium on Signal Processing and Information Technology*, pages 353–356, 2007.

[2] A. Ait-Saïdi, F. Ferraty, R. Kassa, and P. Vieu. Cross-validated estimations in the single-functional index model. *Statistics*, 42(6) :475–494, 2008.

[3] I. Ali, F. Cawkwell, S. Green, and N. Dwyer. Application of statistical and machine learning models for grassland yield estimation based on a hypertemporal satellite remote sensing time series. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 5060–5063, 2014.

[4] U. Amato, A. Antoniadis, and I. De Feis. Dimension reduction in functional regression with applications. *Computational Statistics & Data Analysis*, 50(9) :2422–2446, 2006.

[5] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4 :40–79, 2010.

[6] A. Baíllo, A. Cuevas, and J. Cuesta-Albertos. Supervised classification for a family of Gaussian functional models. *Scandinavian Journal of Statistics*, 38(3) :480–498, 2011.

[7] D. Barbin, G. Elmasry, D.-W. Sun, and P. Allen. Near-infrared hyperspectral imaging for grading and classification of pork. *Meat Science*, 90(1) :259–268, 2012.

[8] E. Ben-Dor, K. Patkin, A. Banin, and A. Karnieli. Mapping of several soil properties using DAIS-7915 hyperspectral scanner data-a case study over clayey soils in Israel. *International Journal of Remote Sensing*, 23(6) :1043–1062, 2002.

[9] I. Ben-Gal, A. Dana, N. Shkolnik, and G. Singer. Efficient construction of decision trees by the dual information distance method. *Quality Technology & Quantitative Management*, 11(1) :133–147, 2014.

[10] A. Berge, A. Jensen, and A. Solberg. Sparse inverse covariance estimates for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 45(5) :1399–1407, 2007.

[11] J. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geoscience and Remote Sensing Magazine*, 1(2) :6–36, 2013.

[12] J. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. Hyperspectral unmixing overview : geometrical, statistical, and sparse regression-based approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2) :354–379, 2012.

[13] J. Boggs, T. Tsegaye, T. Coleman, K. Reddy, and A. Fahsi. Relationship between hyperspectral reflectance, soil nitrate-nitrogen, cotton leaf chlorophyll, and cotton yield : a step toward precision agriculture. *Journal of Sustainable Agriculture*, 22(3) :5–16, 2003.

[14] C. De Boor. *A practical guide to splines*, volume 27. Springer-Verlag New York, 1978.

[15] C. Borggaard and H. Thodberg. Optimal minimal neural interpretation of spectra. *Analytical chemistry*, 64(5) :545–551, 1992.

[16] D. Bosq. *Linear processes in function spaces : theory and applications*, volume 149. Springer Science & Business Media, 2012.

[17] C. Bouveyron and C. Brunet-Saumard. Model-based clustering of high-dimensional data : a review. *Computational Statistics & Data Analysis*, 71 :52–78, 2014.

[18] C. Bouveyron and S. Girard. Robust supervised classification with mixture models : learning from data with uncertain labels. *Pattern Recognition*, 42(11) :2649–2658, 2009.

[19] C. Bouveyron, S. Girard, and C. Schmid. High-Dimensional Discriminant Analysis. *Communications in Statistics - Theory and Methods*, 36(14) :2607–2623, 2007.

[20] L. Breiman. Bagging predictors. *Machine learning*, 24(2) :123–140, 1996.

[21] L. Breiman. Random forests. *Machine learning*, 45(1) :5–32, 2001.

[22] L. Breiman, J. Friedman, C. Stone, and R. Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. CRC press, 1984.

[23] A. Le Bris, N. Chehata, X. Briottet, and N. Paparoditis. Use intermediate results of wrapper band selection methods : a first step toward the optimisation of spectral configuration for land cover classifications. *Proceedings of the IEEE WHISPERS*, 14, 2014.

[24] L. Bruzzone and C. Persello. A novel approach to the selection of spatially invariant features for the classification of hyperspectral images with improved generalization capability. *IEEE Transactions on Geoscience and Remote Sensing*, 47(9) :3180–3191, 2009.

[25] F. Burba, F. Ferraty, and P. Vieu. k-nearest neighbour method in functional nonparametric regression. *Journal of Nonparametric Statistics*, 21(4) :453–469, 2009.

[26] C. Burges. Dimension reduction : a guided tour. *Foundations and Trends in Machine Learning*, 2(4) :275–365, 2010.

[27] P. Burman. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3) :503–514, 1989.

[28] T. Cai and P. Hall. Prediction in functional linear regression. *The Annals of Statistics*, 34(5) :2159–2179, 2006.

[29] G. Camps-Valls and L. Bruzzone. Kernel-based methods for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 43(6) :1351–1362, 2005.

[30] G. Camps-Valls and L. Bruzzone. *Kernel methods for remote sensing data analysis*, volume 2. Wiley Online Library, 2009.

[31] G. Camps-Valls, L. Gomez-Chova, J. Calpe-Maravilla, J. Martin-Guerrero, E. Soria-Olivas, L. Alonso-Chorda, and J. Moreno. Robust support vector method for hyperspectral data classification and knowledge discovery. *IEEE Transactions on Geoscience and Remote Sensing*, 42(7) :1530–1542, 2004.

[32] G. Camps-Valls, J. Mooij, and B. Schölkopf. Remote sensing feature selection by kernel dependence measures. *IEEE Geoscience and Remote Sensing Letters*, 7(3) :587–591, 2010.

[33] H. Cardot, R. Faivre, and M. Goulard. Functional approaches for predicting land use with the temporal evolution of coarse resolution remote sensing data. *Journal of Applied Statistics*, 30(10) :1185–1199, 2003.

[34] H. Cardot, F. Ferraty, and P. Sarda. Spline estimators for the functional linear model. *Statistica Sinica*, 13(3) :571–592, 2003.

[35] H. Cardot, P. Maisongrande, and R. Faivre. Varying-time random effects models for longitudinal data : unmixing and temporal interpolation of remote-sensing data. *Journal of Applied Statistics*, 35(8) :827–846, 2008.

[36] H. Cardot, A. Mas, and P. Sarda. CLT in functional linear regression models. *Probability Theory and Related Fields*, 138(3-4) :325–361, 2007.

[37] R. Carroll, D. Ruppert, L. Stefanski, and C. Crainiceanu. *Measurement error in nonlinear models : a modern perspective*. CRC press, 2nd edition, 2006.

[38] G. Celeux and G. Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5) :781–793, 1995.

[39] J.-W. Chan and D. Paelinckx. Evaluation of Random Forest and Adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery. *Remote Sensing of Environment*, 112(6) :2999–3011, 2008.

[40] C.-C. Chang and C.-J. Lin. LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3) :27, 2011.

[41] C.-I Chang. *Hyperspectral data exploitation : theory and applications*. John Wiley & Sons, 2007.

[42] O. Chapelle, P. Haffner, and V. Vapnik. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5) :1055–1064, 1999.

[43] D. Chen, P. Hall, and H.-G. Müller. Single and multiple index functional regression models with nonparametric link. *The Annals of Statistics*, 39(3) :1720–1747, 2011.

[44] Y. Chen, N. Nasrabadi, and T. Tran. Hyperspectral image classification using dictionary-based sparse representation. *IEEE Transactions on Geoscience and Remote Sensing*, 49(10) :3973–3985, 2011.

[45] Y. Chen, N. Nasrabadi, and T. Tran. Hyperspectral image classification via kernel sparse representation. *IEEE Transactions on Geoscience and Remote Sensing*, 51(1) :217–231, 2013.

[46] X. Cheng, Y. Chen, Y. Tao, C. Wang, M. Kim, and A. Lefcourt. A novel integrated PCA and FLD method on hyperspectral image feature extraction for cucumber chilling damage inspection. *Transactions-American Society of Agricultural Engineers*, 47(4) :1313–1320, 2004.

[47] A. Cheriyadat and L. Bruce. Why principal component analysis is not an appropriate feature extraction method for hyperspectral data. In *IEEE International Geoscience and Remote Sensing Symposium Proceedings*, volume 6, pages 3420–3422, 2003.

[48] J. Chi, H.-C. Kim, and S.-H. Kang. Machine learning-based temporal mixture analysis of hypertemporal Antarctic sea ice data. *Remote Sensing Letters*, 7(2) :190–199, 2016.

[49] B. Chiswick and P. Miller. The international transferability of immigrants' human capital. *Economics of Education Review*, 28(2) :162–169, 2009.

[50] S. Choo and P. Mokhtarian. What type of vehicle do people drive ? The role of attitude and lifestyle in influencing vehicle type choice. *Transportation Research Part A : Policy and Practice*, 38(3) :201–222, 2004.

[51] B. Clarke, E. Fokoue, and H. Zhang. *Principles and theory for data mining and machine learning*. Springer Science & Business Media, 2009.

[52] M. Cochrane. Using vegetation reflectance variability for species level classification of hyperspectral data. *International Journal of Remote Sensing*, 21(10) :2075–2087, 2000.

[53] C. Crambes, A. Kneip, and P. Sarda. Smoothing splines estimators for functional linear regression. *The Annals of Statistics*, 37(1) :35–72, 2009.

[54] A. Cuevas, M. Febrero, and R. Fraiman. Robust estimation and classification for functional data via projection-based depth notions. *Computational Statistics*, 22(3) :481–496, 2007.

[55] C. Davis, J. Bowles, R. Leathers, D. Korwan, T. Downes, W. Snyder, W. Rhea, W. Chen, J. Fisher, W. Bissett, and R. Reisse. Ocean PHILLS hyperspectral imager : design, characterization, and calibration. *Optics Express*, 10(4) :210–221, 2002.

[56] P. Davis, P. Rabinowitz, and W. Rheinbolt. *Methods of numerical integration*. Computer Science and Applied Mathematics. Academic Press, 2nd edition, 2014.

[57] S. Van de Geer. High-dimensional generalized linear models and the lasso. *The Annals of Statistics*, 36(2) :614–645, 2008.

[58] A. Delaigle. Nonparametric kernel methods with errors-in-variables : constructing estimators, computing them, and avoiding common mistakes. *Australian & New Zealand Journal of Statistics*, 56(2) :105–124, 2014.

[59] S. Delalieux, J. Van Aardt, W. Keulemans, E. Schrevens, and P. Coppin. Detection of biotic stress (*venturia inaequalis*) in apple trees using hyperspectral data : non-parametric statistical approaches and physiological implications. *European Journal of Agronomy*, 27(1) :130–143, 2007.

[60] F. Dell'Acqua, P. Gamba, A. Ferrari, J. Palmason, J. Benediktsson, and K. Arnason. Exploiting spectral and spatial information in hyperspectral urban data with high resolution. *IEEE Geoscience and Remote Sensing Letters*, 1(4) :322–326, 2004.

[61] L. Delsol and C. Louchet. Segmentation of hyperspectral images from functional kernel density estimation. *Contributions in infinite-dimensional statistics and related topics*, pages 101–105, 2014.

[62] D. Donoho. High-dimensional data analysis : the curses and blessing of dimensionality. *AMS Mathematical challenges of the 21st century*, 2000.

[63] Q. Du and J. Fowler. Hyperspectral image compression using JPEG2000 and principal component analysis. *IEEE Geoscience and Remote Sensing Letters*, 4(2) :201–205, 2007.

[64] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2) :407–499, 2004.

[65] E. El-Araby, T. El-Ghazawi, J. Le Moigne, and K. Gaj. Wavelet spectral dimension reduction of hyperspectral imagery on a reconfigurable computer. In *Proceedings of the IEEE International Conference on Field-Programmable Technology*, pages 399–402, 2004.

[66] M. Escabias, A. Aguilera, and M. Valderrama. Modeling environmental data by functional principal component logistic regression. *Environmetrics*, 16(1) :95–107, 2005.

[67] G. Evans. *Practical numerical integration*. Wiley New York, 1993.

[68] G. Fan, J. Cao, and J. Wang. Functional data classification for temporal gene expression data with kernel-induced random forests. In *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–5, 2010.

[69] J. Fan and I. Gijbels. *Local polynomial modelling and its applications*, volume 66. CRC Press, 1996.

[70] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR : a library for large linear classification. *The Journal of Machine Learning Research*, 9 :1871–1874, 2008.

[71] R.-E. Fan, P.-H. Chen, and C.-J. Lin. Working set selection using second order information for training support vector machines. *The Journal of Machine Learning Research*, 6 :1889–1918, 2005.

[72] M. Farrell and R. Mersereau. On the impact of PCA dimension reduction for hyperspectral detection of difficult targets. *IEEE Geoscience and Remote Sensing Letters*, 2(2) :192–195, 2005.

[73] F. Fassnacht, C. Neumann, M. Forster, H. Buddenbaum, A. Ghosh, A. Clasen, P. Joshi, and B. Koch. Comparison of feature reduction algorithms for classifying tree species with hyperspectral data on three central European test sites. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6), 2014.

[74] M. Fauvel, J. Benediktsson, J. Chanussot, and J. Sveinsson. Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 46(11) :3804–3814, 2008.

[75] M. Fauvel, J. Chanussot, and J. Benediktsson. Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas. *EURASIP Journal on Advances in Signal Processing*, pages 11–24, 2009.

[76] M. Fauvel, C. Dechesne, A. Zullo, and F. Ferraty. Fast forward feature selection of hyperspectral images for classification with Gaussian mixture models. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6) :2824–2831, 2015.

[77] M. Fauvel, Y. Tarabalka, J. Benediktsson, J. Chanussot, and J. Tilton. Advances in spectral-spatial classification of hyperspectral images. *Proceedings of the IEEE*, 101(3) :652–675, 2013.

[78] M. Fauvel, A. Zullo, and F. Ferraty. Nonlinear parsimonious feature selection for the classification of hyperspectral images. In *6th Workshop on Hyperspectral image and signal processing : evolution in remote sensing (WHISPERS)*, Lausanne, Switzerland, 24-27 June 2014.

[79] F. Ferraty. Regression on functional data : methodological approach with application to near-infrared spectrometry. *Journal de la Société Française de Statistique*, 155(2) :100–120, 2014.

[80] F. Ferraty, A. Goia, E. Salinelli, and P. Vieu. Functional projection pursuit regression. *Test*, 22(2) :293–320, 2013.

[81] F. Ferraty, A. Goia, and P. Vieu. Nonparametric functional methods : new tools for chemometric analysis. In *Statistical methods for biostatistics and related fields*, pages 245–264. Springer, 2007.

[82] F. Ferraty, P. Hall, and P. Vieu. Most-predictive design points for functional data predictors. *Biometrika*, 97(4) :807–824, 2010.

[83] F. Ferraty, A. Mas, and P. Vieu. Nonparametric regression on functional data : inference and practical aspects. *Australian & New Zealand Journal of Statistics*, 49(3) :267–286, 2007.

[84] F. Ferraty and Y. Romain. *The Oxford Handbook of Functional Data Analysis*. Oxford Handbooks in Mathematics. Oxford University Press, 2011.

[85] F. Ferraty and P. Vieu. Curves discrimination : a nonparametric functional approach. *Computational Statistics & Data Analysis*, 44(1) :161–173, 2003.

[86] F. Ferraty and P. Vieu. Functional nonparametric statistics in action. In *The art of semiparametrics*, pages 112–129. Springer, 2006.

[87] F. Ferraty and P. Vieu. *Nonparametric functional data analysis : theory and practice*. New York : Springer-Verlag, 2006.

[88] F. Ferraty, A. Zullo, and M. Fauvel. Nonparametric regression on contamined functional predictor with application to hyperspectral data. *Econometrics and Statistics*, submitted.

[89] L. Ferré and N. Villa. Multilayer perceptron with functional inputs : an inverse regression approach. *Scandinavian Journal of Statistics*, 33(4) :807–823, 2006.

[90] T. Furey, N. Cristianini, N. Duffy, D. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10) :906–914, 2000.

[91] P. Gamba, F. Dell'Acqua, and G. Trianni. Hypertemporal SAR sequences for monitoring land cover dynamics. In *IEEE Radar Conference*, pages 1–5, 2008.

[92] B.-C. Gao, M. Montes, Z. Ahmad, and C. Davis. Atmospheric correction algorithm for hyperspectral remote sensing of ocean color from space. *Applied Optics*, 39(6) :887–896, 2000.

[93] G. Geenens. Curse of dimensionality and related issues in nonparametric functional regression. *Statistics Surveys*, 5 :30–43, 2011.

[94] S. Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350) :320–328, 1975.

[95] A. Ghiyamat and H. Shafri. A review on hyperspectral remote sensing for homogeneous and heterogeneous forest biodiversity assessment. *International Journal of Remote Sensing*, 31(7) :1837–1856, 2010.

[96] P. Gislason, J. Benediktsson, and J. Sveinsson. Random forests for land cover classification. *Pattern Recognition Letters*, 27(4) :294–300, 2006.

[97] A. Goia, C. May, and G. Fusai. Functional clustering and linear regression for peak load forecasting. *International Journal of Forecasting*, 26(4) :700–711, 2010.

[98] C. Gomez, R. Rossel, and A. McBratney. Soil organic carbon prediction by hyperspectral remote sensing and field vis-NIR spectroscopy : an Australian case study. *Geoderma*, 146(3) :403–411, 2008.

[99] E. Greenshtein and J. Park. Application of non parametric empirical Bayes estimation to high dimensional classification. *The Journal of Machine Learning Research*, 10 :1687–1704, 2009.

[100] B. Guo, S. Gunn, R. Damper, and J. Nelson. Customizing kernel functions for SVM-based hyperspectral image classification. *IEEE Transactions on Image Processing*, 17(4) :622–629, 2008.

[101] A. Halimi, Y. Altmann, N. Dobigeon, and J.-Y. Tourneret. Nonlinear unmixing of hyperspectral images using a generalized bilinear model. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11) :4153–4162, 2011.

[102] P. Hall. Principal component for functional data : methodology, theory and discussion. In F. Ferraty and Y. Romain, editors, *The Oxford Handbook of Functional Data Analysis*, Oxford Handbooks in Mathematics, chapter 8, pages 210–234. Oxford University Press, 2011.

[103] P. Hall and J. Horowitz. Methodology and convergence rates for functional linear regression. *The Annals of Statistics*, 35(1) :70–91, 2007.

[104] J. Ham, Y. Chen, M. Crawford, and J. Ghosh. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3) :492–501, 2005.

[105] T. Hastie, A. Buja, and R. Tibshirani. Penalized discriminant analysis. *The Annals of Statistics*, 23(1) :73–102, 1995.

[106] T. Hastie and R. Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1) :155–176, 1996.

[107] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning - data mining, inference and prediction*, volume 2. Springer, 2009.

[108] A. Hoerl and R. Kennard. Ridge regression : biased estimation for nonorthogonal problems. *Technometrics*, 12(1) :55–67, 1970.

[109] J. Hoffbeck and D. Landgrebe. Covariance matrix estimation and classification with limited training data. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 18(7) :763–767, 1996.

[110] T. Hofmann, B. Schölkopf, and A. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3) :1171–1220, 2008.

[111] L. Horváth and P. Kokoszka. *Inference for functional data with applications*. Springer Science & Business Media, 2012.

[112] A. Höskuldsson. PLS regression methods. *Journal of Chemometrics*, 2(3) :211–228, 1988.

[113] T. Hsing and R. Eubank. *Theoretical foundations of functional data analysis, with an introduction to linear operators*. John Wiley & Sons, 2015.

[114] P.-H. Hsu, Y.-H. Tseng, and P. Gong. Dimension reduction of hyperspectral images for classification applications. *Geographic Information Sciences*, 8(1) :1–8, 2002.

[115] G. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, 14(1) :55–63, 1968.

[116] S. Jacquemoud, W. Verhoef, F. Baret, C. Bacour, P. Zarco-Tejada, G. Asner, C. François, and S. Ustin. PROSPECT+SAIL models : a review of use for vegetation characterization. *Remote Sensing of Environment*, 113 :S56–S66, 2009.

[117] G. James. Generalized linear models with functional predictors. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 64(3) :411–432, 2002.

[118] G. James, J. Wang, and J. Zhu. Functional linear regression that's interpretable. *The Annals of Statistics*, 37(5A) :2083–2108, 2009.

[119] A. Jensen, A. Berge, and R. Solberg. Regression approaches to small sample inverse covariance matrix estimation for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 46(10) :2814–2822, 2008.

[120] A. Jensen and A. Solberg. Fast hyperspectral feature reduction using piecewise constant function approximations. *IEEE Geoscience and Remote Sensing Letters*, 4(4) :547–551, 2007.

[121] L. Jimenez and D. Landgrebe. Supervised classification in high-dimensional space : geometrical, statistical, and asymptotical properties of multivariate data. *IEEE Transactions on Systems, Man, and Cybernetics, Part C : Applications and Reviews*, 28(1) :39–54, 1998.

[122] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[123] J. Kalivas. Two data sets of near infrared spectra. *Chemometrics and Intelligent Laboratory Systems*, 37(2) :255–259, 1997.

[124] M. Kästner and T. Villmann. Functional relevance learning in learning vector quantization for hyperspectral data. In *3rd Workshop on Hyperspectral Image and Signal Processing : Evolution in Remote Sensing (WHISPERS)*, pages 1–4, 2011.

[125] H. Kazianka, M. Mulyk, and J. Pilz. A bayesian approach to estimating linear mixtures with unknown covariance structure. *Journal of Applied Statistics*, 38(9) :1801–1817, 2011.

[126] M. Khodadadzadeh, J. Li, A. Plaza, and J. Bioucas-Dias. A subspace-based multinomial logistic regression for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 11(12) :2105–2109, 2014.

[127] W. Kim and M. Crawford. Adaptive classification for hyperspectral image data using manifold regularization kernel machines. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11) :4110–4121, 2010.

[128] U. Kreßel. Pairwise classification and support vector machines. In *Advances in kernel methods*, pages 255–268, 1999.

[129] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink. Sparse multinomial logistic regression : fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6) :957–968, 2005.

[130] B.-C. Kuo, C.-H. Li, and J.-M. Yang. Kernel nonparametric weighted feature extraction for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 47(4) :1139–1155, 2009.

[131] H. Kwon and N. Nasrabadi. Kernel RX-algorithm : a nonlinear anomaly detector for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 43(2) :388–397, 2005.

[132] P. Lagacherie, F. Baret, J.-B. Feret, J. Netto, and J. Robbez-Masson. Estimation of soil clay and calcium carbonate using laboratory, field and airborne hyperspectral measurements. *Remote Sensing of Environment*, 112(3) :825–835, 2008.

[133] D. Landgrebe. *Signal theory methods in multispectral remote sensing.* John Wiley and Sons, New Jersey, 2003.

[134] Y. Lanthier, A. Bannari, D. Haboudane, J. Miller, and N. Tremblay. Hyperspectral data segmentation and classification in precision agriculture : a multi-scale analysis. In *IEEE International Geoscience and Remote Sensing Symposium*, volume 2, pages 585–588, 2008.

[135] R. Lawrence, S. Wood, and R. Sheley. Mapping invasive plants using hyperspectral imagery and Breiman Cutler classifications (RandomForest). *Remote Sensing of Environment*, 100(3) :356–362, 2006.

[136] X. Leng and H.-G. Müller. Classification using functional data analysis for temporal gene expression data. *Bioinformatics*, 22(1) :68–76, 2006.

[137] H. Li, G. Xiao, T. Xia, Y. Tang, and L. Li. Hyperspectral image classification using functional data analysis. *IEEE Transactions on Cybernetics*, 44(9) :1544–1555, 2014.

[138] J. Li, J. Bioucas-Dias, and A. Plaza. Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11) :4085–4098, 2010.

[139] J. Li, J. Bioucas-Dias, and A. Plaza. Hyperspectral image segmentation using a new bayesian approach with active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 49(10) :3947–3960, 2011.

[140] J. Li, J. Bioucas-Dias, and A. Plaza. Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, 50(3) :809–823, 2012.

[141] J. Li, P. Marpu, A. Plaza, J. Bioucas-Dias, and J. Benediktsson. Generalized composite kernel framework for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 51(9) :4816–4829, 2013.

[142] W. Li and Q.-M. Shao. Gaussian processes : inequalities, small ball probabilities and applications. *Handbook of Statistics*, 19 :533–597, 2001.

[143] Y. Li and T. Hsing. On rates of convergence in functional linear regression. *Journal of Multivariate Analysis*, 98(9) :1782–1804, 2007.

[144] F. López-Granados, J. Peña-Barragán, M. Jurado-Expósito, M. Francisco-Fernández, R. Cao, A. Alonso-Betanzos, and O. Fontenla-Romero. Multispectral classification of grass weeds and wheat (Triticum durum) using linear and nonparametric functional discriminant analysis and neural networks. *Weed Research*, 48(1) :28–37, 2008.

[145] M. Lothode, V. Carrere, and R. Marion. Identifying industrial processes through VNIR-SWIR reflectance spectroscopy of their waste materials. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3288–3291, 2014.

[146] L. Ma and W. Cai. Determination of the optimal regularization parameters in hyperspectral tomography. *Applied optics*, 47(23) :4186–4192, 2008.

[147] A. Majumdar, C. Gries, and J. Walker. A non-stationary spatial generalized linear mixed model approach for studying plant diversity. *Journal of Applied Statistics*, 38(9) :1935–1950, 2011.

[148] D. Manolakis, D. Marden, and G. Shaw. Hyperspectral image processing for automatic target detection applications. *Lincoln Laboratory Journal*, 14(1) :79–116, 2003.

[149] D. Manolakis and G. Shaw. Detection algorithms for hyperspectral imaging applications. *IEEE Signal Processing Magazine*, 19(1) :29–43, 2002.

[150] M. De Marchi, R. Riovanto, M. Penasa, and M. Cassandro. At-line prediction of fatty acid profile in chicken breast using near infrared reflectance spectroscopy. *Meat science*, 90(3) :653–657, 2012.

[151] A. Mas and B. Pumo. Functional linear regression with derivatives. *Journal of Nonparametric Statistics*, 21(1) :19–40, 2009.

[152] P. McCullagh and J. Nelder. *Generalized linear models*, volume 2. Chapman and Hall London, 1989.

[153] A. McIntosh, F. Bookstein, J. Haxby, and C. Grady. Spatial pattern analysis of functional brain images using partial least squares. *Neuroimage*, 3(3) :143–157, 1996.

[154] G. McLachlan and D. Peel. *Finite mixture models*. John Wiley & Sons, 2004.

[155] F. Melgani and L. Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8) :1778–1790, 2004.

[156] G. Mercier and M. Lennon. Support vector machines for hyperspectral image classification with spectral-based kernels. In *IEEE International Geoscience and Remote Sensing Symposium*, volume 1, pages 288–290, 2003.

[157] G. Mountrakis, J. Im, and C. Ogole. Support vector machines in remote sensing : a review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3) :247–259, 2011.

[158] H.-G. Müller. Functional modelling and classification of longitudinal data. *Scandinavian Journal of Statistics*, 32(2) :223–240, 2005.

[159] H.-G. Müller and U. Stadtmüller. Generalized functional linear models. *The Annals of Statistics*, 33(2) :774–805, 2005.

[160] E. Nadaraya. On estimating regression. *Theory of Probability and its Applications*, 9 :141–142, 1964.

[161] C. Ordóñez, J. Rodríguez-Pérez, J. Moreira, and E. Sanz. Using hyperspectral spectrometry and functional models to characterize vine-leaf composition. *IEEE Transactions on Geoscience and Remote Sensing*, 51(5) :2610–2618, 2013.

[162] M. Pal. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1) :217–222, 2005.

[163] M. Pal and P. Mather. Support vector machines for classification in remote sensing. *International Journal of Remote Sensing*, 26(5) :1007–1011, 2005.

[164] J. Park, S. Baek, M. Jeong, and S. Bae. Dual features functional support vector machines for fault detection of rechargeable batteries. *IEEE Transactions on Systems, Man, and Cybernetics, Part C : Applications and Reviews*, 39(4) :480–485, 2009.

[165] L. Pasanen and L. Holmström. Bayesian scale space analysis of temporal changes in satellite images. *Journal of Applied Statistics*, 42(1) :50–70, 2015.

[166] T. Pavlenko and D. Von Rosen. Effect of dimensionality on discrimination. *Statistics*, 35(3) :191–213, 2001.

[167] J. Pontius, M. Martin, L. Plourde, and R. Hallett. Ash decline assessment in emerald ash borer-infested regions : a test of tree-level, hyperspectral technologies. *Remote Sensing of Environment*, 112(5) :2665–2676, 2008.

[168] W. Press, S. Teukolsky, and B. Flannery. *Numerical recipes : the art of scientific computing*, chapter 16.1. Gaussian mixture models and k-means clustering. New York : Cambridge University Press, 3 edition, 2007.

[169] J. Qin, T. Burks, M. Kim, K. Chao, and M. Ritenour. Citrus canker detection using hyperspectral reflectance imaging and PCA-based image classification method. *Sensing and Instrumentation for Food Quality and Safety*, 2(3) :168–177, 2008.

[170] R Core Team. *R : a language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

[171] P. Radchenko, X. Qiao, and G. James. Index models for sparsely sampled functional data. *Journal of the American Statistical Association*, 110(510) :824–836, 2015.

[172] J. Ramsay and B. Silverman. *Applied functional data analysis : methods and case studies*, volume 77. Springer, 2002.

[173] J. Ramsay and B. Silverman. *Functional data analysis*. Springer, 2nd edition, 2006.

[174] C. Ramussen and C. Williams. *Gaussian processes for machine learning (adaptive computation and machine learning)*. The MIT Press, Boston, 2006.

[175] F. Ratle, G. Camps-Valls, and J. Weston. Semisupervised neural networks for efficient hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48(5) :2271–2282, 2010.

[176] K. Van Rees, J. Vermunt, and M. Verboord. Cultural classifications under discussion latent class analysis of highbrow and lowbrow reading. *poetics*, 26(5) :349–365, 1999.

[177] P. Reiss and R Ogden. Functional principal component regression and functional partial least squares. *Journal of the American Statistical Association*, 102(479) :984–996, 2007.

[178] S. Roessner, K. Segl, U. Heiden, and H. Kaufmann. Automated differentiation of urban surfaces based on airborne hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 39(7) :1525–1532, 2001.

[179] F. Rossi and N. Villa. Support vector machine for functional data classification. *Neurocomputing*, 69(7) :730–742, 2006.

[180] P. Rosso, S. Ustin, and A. Hastings. Mapping marshland vegetation of San Francisco Bay, California, using hyperspectral data. *International Journal of Remote Sensing*, 26(23) :5169–5191, 2005.

[181] T. Schmid, M. Koch, and J. Gumuzzio. Multisensor approach to determine changes of wetland characteristics in semiarid environments (central Spain). *IEEE Transactions on Geoscience and Remote Sensing*, 43(11) :2516–2525, 2005.

[182] L. Schumaker. *Spline functions : basic theory*. Cambridge University Press, 3rd edition, 2007.

[183] D. Schwarz, I. König, and A. Ziegler. On safari to Random Jungle : a fast implementation of Random Forests for high-dimensional data. *Bioinformatics*, 26(14) :1752–1758, 2010.

[184] S. Serpico and L. Bruzzone. A new search algorithm for feature selection in hyperspectral remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 39(7) :1360–1367, 2001.

[185] S. Serpico and G. Moser. Extraction of spectral channels from hyperspectral images for classification purposes. *IEEE Transactions on Geoscience and Remote Sensing*, 45(2) :484–495, 2007.

[186] H. Shang. A bayesian approach for determining the optimal semi-metric and bandwidth in scalar-on-function quantile regression with unknown error density and dependent functional data. *Journal of Multivariate Analysis*, 146 :95–104, 2016.

[187] G. Shaw and D. Manolakis. Signal processing for hyperspectral image exploitation. *IEEE Signal Processing Magazine*, 19(1) :12–16, 2002.

[188] N. Sinelli, S. Limbo, L. Torri, V. Di Egidio, and E. Casiraghi. Evaluation of freshness decay of minced beef stored in high-oxygen modified atmosphere packaged at different temperatures using nir and mir spectroscopy. *Meat science*, 86(3) :748–752, 2010.

[189] J. Solomon and B. Rock. Imaging spectrometry for earth remote sensing. *Science*, 228(4704) :1147–1152, 1985.

[190] A. Statnikov, L. Wang, and C. Aliferis. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC bioinformatics*, 9(1) :319, 2008.

[191] D. Stein, S. Beaven, L. Hoff, E. Winter, A. Schaum, and A. Stocker. Anomaly detection from hyperspectral imagery. *IEEE Signal Processing Magazine*, 19(1) :58–69, 2002.

[192] L. Stratton, D. O'Toole, and J. Wetzel. A multinomial logit model of college stopout and dropout behavior. *Economics of Education Review*, 27(3) :319–331, 2008.

[193] S. Sugianto and S. Laffan. Functional data analysis of multi-angular hyperspectral data on vegetation. *Aceh International Journal of Science and Technology*, 1(1), 2012.

[194] Y. Tarabalka, J. Benediktsson, and J. Chanussot. Spectral-spatial classification of hyperspectral imagery based on partitional clustering techniques. *IEEE Transactions on Geoscience and Remote Sensing*, 47(8) :2973–2987, 2009.

[195] Y. Tarabalka, J. Benediktsson, J. Chanussot, and J. Tilton. Multiple spectral-spatial classification approach for hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11) :4122–4132, 2010.

[196] Y. Tarabalka, J. Chanussot, and J. Benediktsson. Segmentation and classification of hyperspectral images using watershed transformation. *Pattern Recognition*, 43(7) :2367–2379, 2010.

[197] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. Benediktsson. SVM- and MRF-based method for accurate classification of hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 7(4) :736–740, 2010.

[198] J. Tarrío-Saavedra, S. Naya, M. Francisco-Fernández, J. López-Beceiro, and R. Artiaga. Functional nonparametric classification of wood species from thermal data. *Journal of thermal analysis and calorimetry*, 104(1) :87–100, 2011.

[199] P. Thenkabail, J. Lyon, and A. Huete. *Hyperspectral remote sensing of vegetation*. CRC Press, 2011.

[200] M. Thoma. Electrical energy usage over the business cycle. *Energy Economics*, 26(3) :463–485, 2004.

[201] T. Tian and G. James. Interpretable dimension reduction for classifying functional data. *Computational Statistics & Data Analysis*, 57(1) :282–296, 2013.

[202] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[203] C. Timmermans, L. Delsol, and R. Von Sachs. Using Bagidis in nonparametric functional data analysis : predicting from curves with sharp local features. *Journal of Multivariate Analysis*, 115 :421–444, 2013.

[204] L. Toloşi and T. Lengauer. Classification with correlated features : unreliability of feature ranking and solutions. *Bioinformatics*, 27(14) :1986–1994, 2011.

[205] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2 :45–66, 2002.

[206] D. Tuia, F. Pacifici, M. Kanevski, and W. Emery. Classification of very high spatial resolution imagery using mathematical morphology and support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 47(11) :3866–3879, 2009.

[207] D. Tuia, M. Volpi, M. Dalla Mura, A. Rakotomamonjy, and R. Flamary. Automatic feature learning for spatio-spectral image classification with sparse SVM. *IEEE Transactions on Geoscience and Remote Sensing*, 52(10) :6062–6074, 2014.

[208] T. Udelhoven and M. Stellmes. Changes in land surface conditions on the Iberian Peninsula (1989 to 2004) detected by means of time series analysis from hypertemporal remote sensing data. In *International Workshop on the Analysis of Multi-temporal Remote Sensing Images*, pages 1–6, 2007.

[209] V. Vapnik. *The nature of statistical learning theory*. Information Science and Statistics. Springer, 1996.

[210] A. Villa. *Advanced spectral unmixing and classification methods for hyperspectral remote sensing data*. PhD thesis, Université de Grenoble ; 102 Univ of Iceland, Reykjavik, 2011.

[211] A. Villa, J. Benediktsson, J. Chanussot, and C. Jutten. Hyperspectral image classification with independent component discriminant analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 49(12) :4865–4876, 2011.

[212] V. Vinzi, W. Chin, J. Henseler, and H. Wang. *Handbook of partial least squares : concepts, methods and applications*. Springer Handbooks of Computational Statistics. Springer Berlin Heidelberg, 2010.

[213] G. Watson. Smooth regression analysis. *Sankhya : The Indian Journal of Statistics*, 26 :359–372, 1964.

[214] H. Wold. Estimation of principal components and related models. *Multivariate Analysis*, pages 391–420, 1966.

[215] Y. Wu, J. Fan, and H.-G. Müller. Varying-coefficient functional linear regression. *Bernoulli*, 16(3) :730–758, 2010.

[216] F. Yao, E. Lei, and Y. Wu. Effective dimension reduction for sparse functional data. *Biometrika*, 102(2) :421–437, 2015.

[217] F. Yao, H.-G. Müller, and J.-L. Wang. Functional linear regression analysis for longitudinal data. *The Annals of Statistics*, 33(6) :2873–2903, 2005.

[218] Y. Zhao, J. Yang, Q. Zhang, L. Song, Y. Cheng, and Q. Pan. Hyperspectral imagery super-resolution by sparse representation and spectral regularization. *EURASIP Journal on Advances in Signal Processing*, (1) :1–10, 2011.

[219] L. Zhou, H. Wu, J. Li, Z. Wang, and L. Zhang. Determination of fatty acids in broiler breast meat by near-infrared reflectance spectroscopy. *Meat science*, 90(3) :658–664, 2012.

[220] A. Zullo, F. Fauvel, and F. Ferraty. Sélection de variables pour l'imagerie hyperspectrale. In *46e Journées de Statistique, Société Française de Statistique*, Rennes, 2-6 juin 2014.

[221] A. Zullo, M. Fauvel, and F. Ferraty. Classification d'images hyperspectrales par des méthodes fonctionnelles non-paramétriques. In *3ème colloque scientifique du Groupe Hyperspectral de la Société Française de Photogrammétrie et de Télédétection*, Porquerolles, 15-16 mai 2014.

[222] A. Zullo, M. Fauvel, and F. Ferraty. Comparison of functional and multivariate spectral-based supervised classification methods in hyperspectral image. *Journal of Applied Statistics*, submitted.

[223] A. Zullo, M. Fauvel, F. Ferraty, M. Goulard, and P. Vieu. Non-parametric functional methods for hyperspectral image classification. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 3422–3425, Quebec city, July 13th-18th, 2014.

[224] A. Zullo, F. Ferraty, and M. Fauvel. Débruitage d'images hyperspectrales avec un modèle de bruit hétéroscédastique : application à l'estimation de variables biophysiques par régression non-paramétrique fonctionnelle. In *4ème colloque scientifique du Groupe Hyperspectral de la Société Française de Photogrammétrie et de Télédétection*, Grenoble, 11-13 mai 2016.

Résumé. En imagerie hyperspectrale, chaque pixel est associé à un spectre provenant de la réflectance observée en d points de mesure (i.e., longueurs d'onde). On se retrouve souvent dans une situation où la taille d'échantillon n est relativement faible devant le nombre d de variables. Ce phénomène appelé «fléau de la dimension» est bien connu en statistique multivariée. Plus d augmente devant n, plus les performances des méthodologies statistiques standard se dégradent. Les spectres de réflectance intègrent dans leur dimension spectrale un continuum qui leur confère une nature fonctionnelle. Un hyperspectre peut être modélisé par une fonction univariée de la longueur d'onde, sa représentation produisant une courbe. L'utilisation de méthodes fonctionnelles sur de telles données permet de prendre en compte des aspects fonctionnels tels que la continuité, l'ordre des bandes spectrales, et de s'affranchir des fortes corrélations liées à la finesse de la grille de discrétisation. L'objectif principal de cette thèse est d'évaluer la pertinence de l'approche fonctionnelle dans le domaine de la télédétection hyperspectrale lors de l'analyse statistique. Nous nous sommes focalisés sur le modèle non-paramétrique de régression fonctionnelle, couvrant la classification supervisée. Dans un premier temps, l'approche fonctionnelle a été comparée avec des méthodes multivariées usuellement employées en télédétection. L'approche fonctionnelle surpasse les méthodes multivariées dans des situations délicates où l'on dispose d'une petite taille d'échantillon d'apprentissage combinée à des classes relativement homogènes (c'est-à-dire difficiles à discriminer). Dans un second temps, une alternative à l'approche fonctionnelle pour s'affranchir du fléau de la dimension a été développée à l'aide d'un modèle parcimonieux. Ce dernier permet, à travers la sélection d'un petit nombre de points de mesure, de réduire la dimensionnalité du problème tout en augmentant l'interprétabilité des résultats. Dans un troisième temps, nous nous sommes intéressés à la situation pratique quasi-systématique où l'on dispose de données fonctionnelles contaminées. Nous avons démontré que pour une taille d'échantillon fixée, plus la discrétisation est fine, meilleure sera la prédiction. Autrement dit, plus d est grand devant n, plus la méthode statistique fonctionnelle développée est performante.

Abstract. In hyperspectral imaging, each pixel is associated with a spectrum derived from observed reflectance in d measurement points (i.e., wavelengths). We are often facing a situation where the sample size n is relatively low compared to the number d of variables. This phenomenon called "curse of dimensionality" is well known in multivariate statistics. The more d increases with respect to n, the more standard statistical methodologies performances are degraded. Reflectance spectra incorporate in their spectral dimension a continuum that gives them a functional nature. A hyperspectrum can be modelised by an univariate function of wavelength and his representation produces a curve. The use of functional methods allows to take into account functional aspects such as continuity, spectral bands order, and to overcome strong correlations coming from the discretization grid fineness. The main aim of this thesis is to assess the relevance of the functional approach in the field of hyperspectral remote sensing for statistical analysis. We focused on the nonparametric fonctional regression model, including supervised classification. Firstly, the functional approach has been compared with multivariate methods usually involved in remote sensing. The functional approach outperforms multivariate methods in critical situations where one has a small training sample size combined with relatively homogeneous classes (that is to say, hard to discriminate). Secondly, an alternative to the functional approach to overcome the curse of dimensionality has been proposed using parsimonious models. This latter allows, through the selection of few measurement points, to reduce problem dimensionality while increasing results interpretability. Finally, we were interested in the almost systematic situation where one has contaminated functional data. We proved that for a fixed sample size, the finer the discretization, the better the prediction. In other words, the larger d is compared to n, the more effective the functional statistical method is.